中国研究生创新实践系列大赛

“华为杯”第十七届中国研究生数学建模竞赛

学　　校	华中科技大学
参赛队号	20104870036
队员姓名	1. 韩冬
	2. 张韵
	3. 李望铭

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

题 目　降低汽油精制过程中的辛烷值损失模型

摘　　要：

现有技术在对汽油进行催化裂化过程中，普遍降低了汽油辛烷值，故对汽油精制过程进行有效建模，具有重要的研究价值。本文从理想到实际，逐步按需建立了降低汽油精制过程中的辛烷值损失模型，对题目所给出的问题进行快速有效求解。

针对问题一，对 285 号和 313 号数据样本的原始数据进行处理。首先，结合汽油精制工业背景对原始数据和样本数据进行分析，提出四种适用于本文的数据**预处理方法**：针对采集时间异常的样本进行剔除处理；针对超出取值范围的数据进行剔除；针对异常非操作变量数据采用**依拉达准则**进行剔除；针对缺失数据采用**拉格朗日插值**和**均值插补**进行填充。最后，对预处理后的附件三中 285 号和 313 号原始数据样本进行计算，并将处理后的结果分别加入到附件一中相应位置。

针对问题二，筛选出辛烷值损失模型中的主要建模变量。首先，根据变量类型不同对操作变量和非操作变量进行差异化筛选建模：对于操作变量，建立**基于随机森林的特征重要度筛选模型**；对于非操作变量，建立**基于相关性分析的非操作变量特征筛选模型**。然后，对模型进行求解，从所有变量中筛选出 26 个建模主要变量，其中包括 16 个主要操作变量和 10 个主要非操作变量。最后，采用**信息熵理论**对主要变量的代表性进行验证，采用**相关性分析法**对主要变量的独立性进行验证。

针对问题三，建立辛烷值（RON）损失预测模型。首先，分析得出辛烷值损失由“原料辛烷值”与“产品辛烷值”经二次计算得到，直接建模预测辛烷值损失则效果不佳，考虑进行**预测目标转换**，先建立产品辛烷值的预测模型，再根据结果得到相应的辛烷值损失值。其次，分别建立基于**多元线性回归、随机森林、梯度提升回归的产品辛烷值预测模型**，并对三种模型的预测效果进行分析。最后，对模型进行**敏感性分析**，探究代表变量与辛烷值损失之间的内在联系。

针对问题四，对主要变量操作方案进行优化。首先，以最大化辛烷值损失降幅为目标函数，以 16 个主要操作变量为决策变量，以产品硫含量和决策变量取值范围为约束，建立主要变量操作方案优化的**线性规划模型**。其次，采用**自适应和声搜索算法**对模型进行求解，并对算法关键内容进行针对性设计，求解得到辛烷值损失降幅大于 30%的样本占比为 71.7%，并对这部分样本的主要操作方案进行优化和分析。最后，进行算法对比分析，证明了自适应和声搜索算法的有效性和优越性。

针对问题五，对模型进行可视化展示。考虑到问题特性和汽油精制过程的复杂性，拟从三个层面进行本文模型的可视化展示：一、产品性质层；二、操作方案层；三、内在关联层。并以 133 号样本为展示对象，显示出相应的可视化图像。

最后，对问题的模型与算法中的优点和不足进行了总结。

关键词：数据挖掘；随机森林；相关分析；GBR 模型；线性规划；和声搜索算法；可视化

目录

1. 问题背景与问题重述

1.1. 问题背景

随着人民物质生活的不断提高，汽车数量也在不断的增加。汽车排放的尾气是大气主要污染源，因此汽车尾气排放带来的环保问题越来越收到世界各国的广泛关注，世界各国也制定了日益严格的汽油质量标准。为此，降低汽油硫、烯烃含量是减少汽车污染物排放的有效手段之一。

辛烷值（以 RON 表示）是反映汽油燃烧性能最重要的指标。现有的催化裂化精制汽油脱硫装置虽然能够有效的脱除汽油中的硫化物，但是在脱硫的同时会使大量烯烃饱和，从而导致汽油的辛烷值急剧下降。辛烷值每降低 1 个单位，相当于损失约 150 元/吨。因此，降低汽油辛烷值损失对石化企业而言是至关重要的。

为了预测汽油辛烷值的损失，一般是通过数据关联或机理建模的方法来实现。而由于炼油工艺过程复杂、设备多样，且其操作变量之间具有高度非线性和相互强藕联的关系，因此采用数据关联或机理建模等传统的手段对化工过程进行建模存在一定的弊端。近年来，数据挖掘技术引起了产业界极大的关注，其主要原因是能够从大量的数据中揭示隐含的、未知的并具有潜在价值的信息，并提供给人们做决策。因此，采用数据挖掘技术预测辛烷值损失，能够有效地避免传统数据关联模型中变量少，机理建模对原料分析要求高等问题，同时能够对优化过程及时响应，有效地降低精制汽油的辛烷值损失，提高装置的经济效益。

1.2. 问题重述

基于上述研究背景，本文需要研究解决以下问题：

问题 1：数据预处理

请参考近 4 年的工业数据(见附件一“325 个数据样本数据.xlsx”)的预处理结果，依“样本确定方法”（附件二）对 285 号和 313 号数据样本进行预处理（原始数据见附件三“285 号和 313 号样本原始数据.xlsx”）并将处理后的数据分别加入到附件一中相应的样本号中，供下面研究使用。

问题 2：寻找建模主要变量

由于催化裂化汽油精制过程是连续的，虽然操作变量每 3 分钟就采样一次，但辛烷值（因变量）的测量比较麻烦，一周仅 2 次无法对应。但根据实际情况可以认为辛烷值的测量值是测量时刻前两小时内操作变量的综合效果，因此预处理中取操作变量两小时内的平均值与辛烷值的测量值对应。这样产生了 325 个样本（见附件一）。

建立降低辛烷值损失模型涉及包括 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、2 个产品性质等变量以及另外 354 个操作变量（共计 367 个变量），工程技术应用中经常使用先降维后建模的方法，这有利于忽略次要因素，发现并分析影响模型的主要变量与因素。因此，请你们根据提供的 325 个样本数据（见附件一），通过降维的方法从 367 个操作变量中筛选出建模主要变量，使之尽可能具有代表性、独立性（为了工程应用方便，建议降维后的主要变量在 30 个以下），并请详细说明建模主要变量的筛选过程及其合理性。（提示：请考虑将原料的辛烷值作为建模变量之一）。

问题 3：建议辛烷值损失预测模型

采用上述样本和建模主要变量，通过数据挖掘技术建立辛烷值（RON）损失预测模型，并进行模型验证。

问题 4：主要变量操作方案的优化

要求在保证产品硫含量不大于 5 μg/g 的前提下，利用你们的模型获得 325 个数据样本(见附件四“325 个数据样本数据.xlsx”)中，辛烷值（RON）损失降幅大于 30%的样本对应的主要变量优化后的操作条件（优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，以它们在样本中的数据为准）。

问题 5：模型的可视化展示

工业装置为了平稳生产，优化后的主要操作变量（即：问题 2 中的主要变量）往往只能逐步调整到位，请你们对 133 号样本（原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，以样本中的数据为准），以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹。（各主要操作变量每次允许调整幅度值 Δ 见附件四“354 个操作变量信息.xlsx”）。

2. 模型假设与符号系统

2.1. 模型假设

根据题目中所给信息以及要求，本文做出如下假设：

(1) 假设附件所提供的数据绝大部分是真实可靠的；
(2) 假设采集数据不受催化裂化汽油精制装置的磨损、老化影响；
(3) 假设可以应用附件四中各操作变量取值范围信息对数据进行预处理。

2.2. 符号系统

表 2-1 符号系统

序号	符号	含义
1	VS	解释回归模型的方差得分
2	MAE	平均绝对误差
3	MSE	均方误差
4	r^2	判定系数
5	ΔRON	优化操作变量之前的辛烷值损失
6	$\Delta RON^{'}$	优化操作变量之后的辛烷值损失
7	$x_i\,(i=1,2,\ldots 16)$	主要操作变量
8	$S^{'}$	优化操作变量之后的产品硫含量

9	$x_{i\min}$	主要操作变量可取最小值
10	$x_{i\max}$	主要操作变量可取最大值

注：其他符号在文中说明。

3. 问题一：样本及原始数据处理方法研究

3.1. 问题分析

针对问题一，要求对题目所提供的数据进行处理。首先，对原始数据和样本数据进行分析：附件一提供了 325 个样本数据，包含近 4 年的工业采集数据样本；附件三提供了 285 和 313 号样本两小时内采集的原始数据。其中，原始数据可能在采集中或传输过程中因种种原因出现数据缺失、数据异常等问题，导致数据质量下降。这种情况下，如果直接使用这些不良数据进行后续汽油精制过程中的辛烷值损失建模，得到的结果也不具有说服性。因此，考虑在后续建模分析前，对样本数据和原始数据进行预处理。

通过对提供的工业细则（附件二）进行研究提炼，本文对上述数据的预处理包含如下过程：首先，针对采集时间异常的数据进行删除处理；其次，将超出各操作变量的取值范围的数据进行剔除；然后，采用依拉达准则对异常的非操作变量数据进行剔除；最后，采用拉格朗日插值或均值插补的方法对附件中的缺失数据进行补充。

最后，按照要求对经上述方法预处理后的附件三中 285 号和 313 号原始数据样本进行计算，并将处理后的结果分别加入到附件 1 中相应位置。问题一的思路流程图如图 3-1 所示：

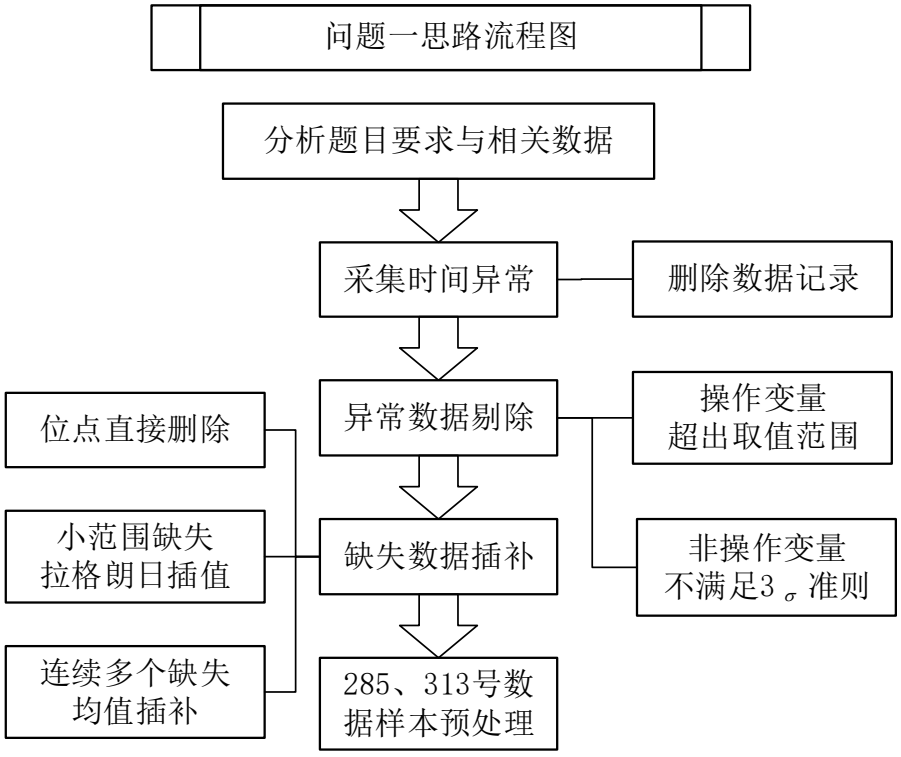

图 3-1 问题一思路流程图

3.2. 异常数据预处理

异常数据可能有多个来源，如数据本身、数据存储过程或者数据转换过程。由于异常数据会影响特征，也会影响最后的模型结果，因此对数据进行预处理十分必要。异常值的处理方法有多种，如删除记录、视为缺失值、平均值修正、不处理等。异常值如何处理，需要视具体应用背景分析而定。

3.2.1. 异常采集时间处理

对附件一中各样本采集时间数据进行分析，统计不同样本的采集时间如表 3-1 所示，可见，大多数样本的采集时刻在上午 8 时附近。由题目得，部分操作变量体现温度、压强等物理性质，同时部分非操作变量也存在潜在的化学联系，因此本文界定数据采集时刻差异大的样本为后续建模分析中的“干扰样本”，需予以剔除。同时，查阅相关资料确定剔除“分隔点”为 12 时(该时刻前后环境因素相差较大)，将附件一中采集时刻在 12 时之后的 3 条样本数据进行剔除。

表 3-1 不同采集时刻的样本数量表

采集时刻	8 时	9 时	10 时	12 时	13 时	14 时	20 时
样本数量	316	2	3	1	1	1	1

3.2.2. 异常值剔除

通过采集得到的原始数据通常会存在少数异常值，并且异常值的存在也是样本数据分布的常态。异常值又被定义为异常或噪声，包括处于特定分布区域或分布范围之外的数据。异常数据通常分为两种：伪异常和真异常。其中，伪异常是指由于特定的动作产生，是正常反应业务的状态，而不是数据本身的异常；真异常则不是由特定动作产生，而是数据本身分布异常，即离群点。

结合上述分析，对样本数据(附件一)和原始数据(附件三)进行异常值剔除，方法如下：

(1) 剔除范围之外的数据

根据附件二中数据整定方法和附件四给出的 354 个操作变量的取值范围信息，将样本数据和原始数据中各操作变量的样本取值超出对应取值范围的位点值视为异常数据并予以剔除。

附件一部分异常示例如表 3-2 所示，S-ZORB.AT_5201.PV(精制汽油出装置硫含量)的取值范围为 0-5，S-ZORB.SIS_LT_1001.PV(原料缓冲罐液位)的取值范围为 40-80，应对相应超出范围的数据进行剔除。同理对附件三中原始样本数据进行处理。

表 3-2 附件一中部分异常位点示例一

样本编号	时间	S-ZORB.CAL_H2.PV	…	S-ZORB.AT_5201.PV	…	S-ZORB.SIS_LT_1001.PV	…
		氢油比	…	精制汽油出装置硫含量	…	原料缓冲罐液位	…
19	2020/3/25 8:24:26	0.280241965	…	-0.32023025	…	190.5452	…

13	2020/4/16 8:00:00	0.28865038	…	1.026460555	…	190.5452	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(2) 剔除处于特定分布区域的数据

上述模式主要适用于操作变量中异常数据的剔除。考虑变量本身的特性，非操作变量的异常多属于数据本身分布的异常，即离群点。因此，本文采用拉依达准则对非操作变量进行异常数据剔除。

3σ准则（拉依达准则）：设对被测量变量进行等精度测量，得到 x_1，x_2，……，x_n，算出其算术平均值 x 及剩余误差 $v_i=x_i-x$（i=1,2,...,n），并按贝塞尔公式算出标准误差 σ，若某个测量值 x_b 的剩余误差 v_b（1<=b<=n），满足$|v_b|=|x_b-x|>3\sigma$，则认为 x_b 是含有粗大误差值的坏值，应予剔除。贝塞尔公式如下：

$$\sigma=\left[\frac{1}{n-1}\sum_{i=1}^{n}v_i^2\right]^{1/2}=\left\{\left(\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\right)\Big/n\Big/n-1\right\}^{1/2}$$

在上述模式下，将与平均值偏差的绝对值超出 3 倍标准差的数据视为异常数据，并予以剔除。附件一中超出拉依达准则的部分样本数据示例如表 3-3 所示，。

表 3-3 附件一中部分异常位点示例二

样本编号	时间	原料性质				产品性质	...
		辛烷值 RON	…	芳烃 V%	…	辛烷值 RON	...
14	2020/4/14 8:00:00	86.8	…	16.00	…	85.4	...
142	2019/1/18 8:00:00	85.3	…	23.9	…	85.1	...
⋮	⋮	⋮	⋮	⋮	⋮		

3.2.3. 缺失值处理

多数情况下，数据采集过程中会存在数据缺失的现象。造成该现象的原因可能是采集装置未能实时感应到数据的变化，或由于通讯设备故障导致数据上传失败。统计学中将数据缺失记录称为不完全观测，这类数据缺失会对模型结果产生较大的影响。

数据缺失的处理方法通常包括：一、删除特征变量：若某一特征变量中存在大量缺失值（缺失量占总样本量 30%以上），则有理由认定该特征提供的信息有限，可选择删除这一特征；二、删除样本：若整个数据集缺失值较少或者所占总数据量极少，可以直接删除含有缺失值的样本记录；三、数据插补：对缺失数据进行统计学补充，包括统计量填充、插值法填充等。

本文考虑的数据缺失情况包括：原始数据和样本数据本身的缺失；经过异常数据剔除后的数据空缺。结合上述分析，对样本数据（附件一）和原始数据（附件三）中的数据进行缺失值处理：

(1) 直接删除缺失值

由于附件一和附件三中的样本总量较少，若直接删除含有缺失值的样本记录会导致后续建模样本数量不足。同时，样本数据位点（操作变量）数量相对较多，并且部分位点与产品性质间关系较弱（6.4.2 节中说明），所以考虑采用直接删除的方法，将样本观测值缺失量大于 30%的位点删除。综合上述，需删除的位点信息如表 3-4 所示。

表 3-4 附件一和附件三中缺失量大于 30%的位点

操作变量	S-ZORB.AT_5201.PV	S-ZORB.SIS_LT_1001.PV	S-ZORB.AI_2903.PV	S-ZORB.FT_1204.TOTAL
	精制汽油出装置硫含量	原料缓冲罐液位	再生烟气氧含量	
附件 1 缺失量	45.2%	100%	96.9%	42.2%
附件 3-1 缺失量	0%	100%	100%	100%
附件 3-2 缺失量	97.5%	100%	100%	100%

(2) 缺失值插补

拉格朗日插值法(Lagrange interpolation)是一种多项式插值方法。如对实践中某个物理量进行观测，在若干个不同的地方得到相应的观测值，拉格朗日插值法可以找到一个多项式，其恰好在各个观测点取到观测到的值。从数学上来讲，拉格朗日插值法可以给出一个恰好穿过二维平面上若干个已知点的多项式函数，并且可以证明，经过 n+1 个互异的点的次数不超过 n 的多项式是唯一存在的。应用拉格朗日插值公式所得到的拉格朗日插值多项式为：

$$L_n(x)=\sum_{j=0}^{n} y_j l_j(x) \tag{3.1}$$

其中，每个$l_j(x)$为拉格朗日基本多项式（或称为插值基函数），其表达式为：

$$l_j(x)=\prod_{i=0,i\neq j}^{n}\frac{x-x_i}{x-x_j} \tag{3.2}$$

在插值计算中，为了减少截断误差，选择插值节点时尽量选取与插值点距离较近的一些节点。本文针对连续缺失小于 3 条的数据，取其前后 5 条数据进行拉格朗日插值从而将缺失部分补充完整；针对连续缺失条数较多的数据(如附件三中样本 313)，由于无法在其相邻时刻取到足够多的点，因此采用均值插补的方法将剩余缺失数据补充完整。插补的部分结果示例如图 3-2 所示。

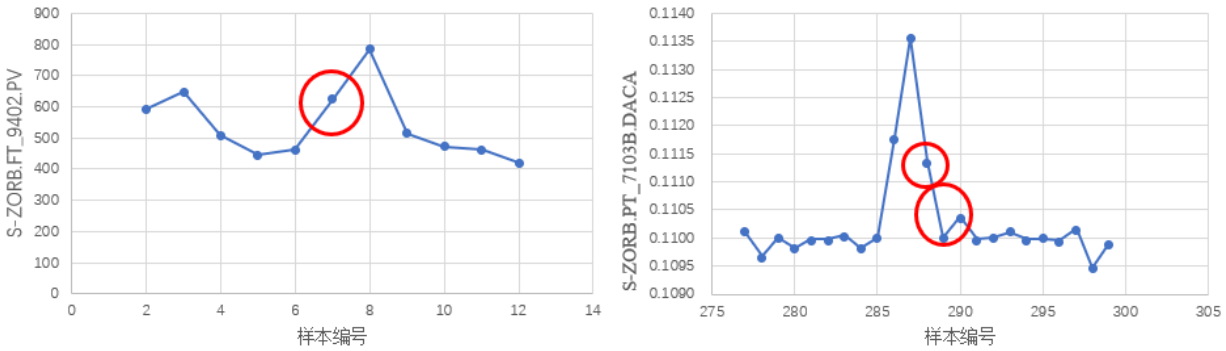

图 3-2 附件一(右)和附件三(左)数据插补部分结果示例图

3.2.4. 数据预处理结果

根据上述方法和步骤，对样本数据（附件一）和原始数据（附件三）进行预处理，处理结果如图 3-3 所示。

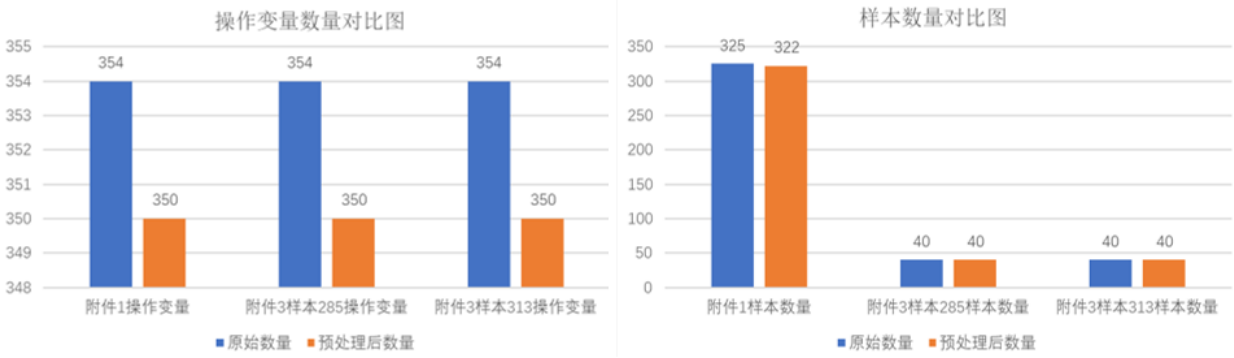

图 3-3 数据预处理前后操作变量及样本数量对比图

3.3. 285 号和 313 号数据样本处理结果

利用预处理后的附件三数据，结合附件二中“样本确定方法”，以辛烷值数据测定的时间点为基准时间，取其前 2 个小时的操作变量数据的平均值作为对应辛烷值的操作变量数据。表 3-4 给出 285 号样本和 313 号样本的部分示例数据。图 3-4 给出 285 号样本与 313 号样本部分数据预处理前后的对比图。

结合数据预处理结果以及图 3-4 所示的部分数据对比可知，采用本文所描述的数据预处理过程得到的结果与附件一中的数据差别较小，个别数据存在较明显差别。

表 3-4 预处理后 285 号样本和 313 号样本部分示例数据表

样本编号	时间	S-ZORB.CAL_H2.PV	…	S-ZORB.PT_2501.DACA	…	S-ZORB.AT-0013.DACA.PV	…
		氢油比	…	D-107 顶压力	…	S_ZORB AT-0013	…
285	2017/7/17 8:00:00	0.2734	…	0.16904	…	0.38156	…
313	2017/5/15 8:00:00	0.26194	…	0.14035	…	0.40229	…

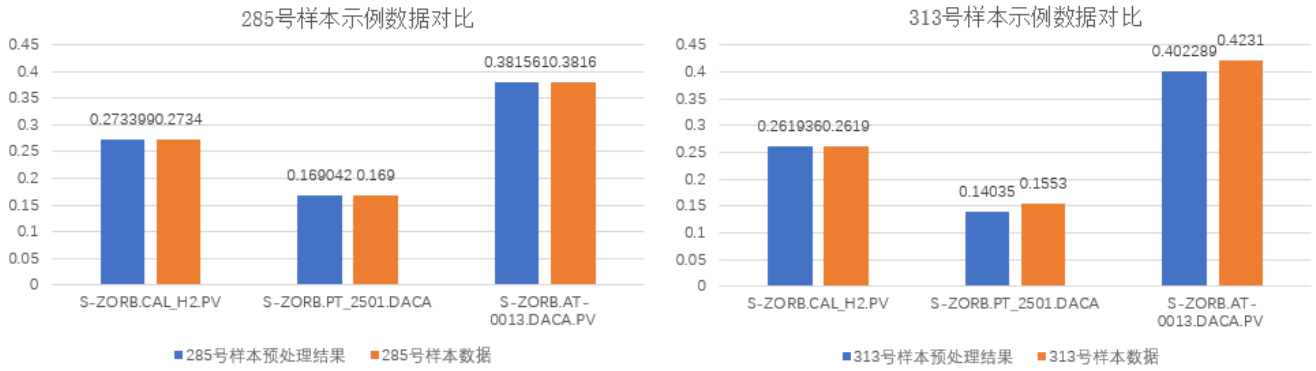

图 3-4 285 号与 313 号样本示例数据对比图

3.4. 结果结论

通过问题一中所描述的数据预处理方法，分别求解得到 285 号和 313 号样本数据预处理之后的结果。与附件 1 中原始样本数据进行对比发现，绝大多数的数据几乎没有差别，而存在极个别数据稍有偏差。初步分析，个别偏差较大的数据对应的操作变量与温度、压强等性质存在密切联系，因此我们认为由于本文中考虑到采集时间异常对采集数据也会造成一定的影响，并对这类异常数据进行剔除，所以得到的结果较附件 1 中所给出的数据更为合理和准确。

4. 问题二：辛烷值损失模型主要变量筛选方法研究

4.1. 问题二分析

针对问题二，要求筛选出辛烷值损失模型中的主要建模变量，由于不同类型变量（原料性质、待生吸附剂性质、再生吸附剂性质）之间含义与性质相差较大，考虑将其分开考虑。对于操作变量（控制变量），题目表明其具有高度非线性和相互强耦联的关系，所以传统线性降维方法（如 PCA 等）不再适用，本文使用基于随机森林的特征重要度筛选方法，对 354 个操作变量进行有效筛选和降维；对于原料性质和吸附剂性质（待生吸附剂、再生吸附剂）变量，由于其具有潜在的化学相关性，本文对其进行相关性分析，保留相互间相关性较弱的变量，甄选相关性较强的变量。最后，对建模主要变量的筛选结果进行合理性分析，通过信息熵理论验证了变量筛选后主要变量的代表性，通过相关性分析验证了变量筛选后主要变量的独立性。图 4-1 给出该问题的思路流程图。

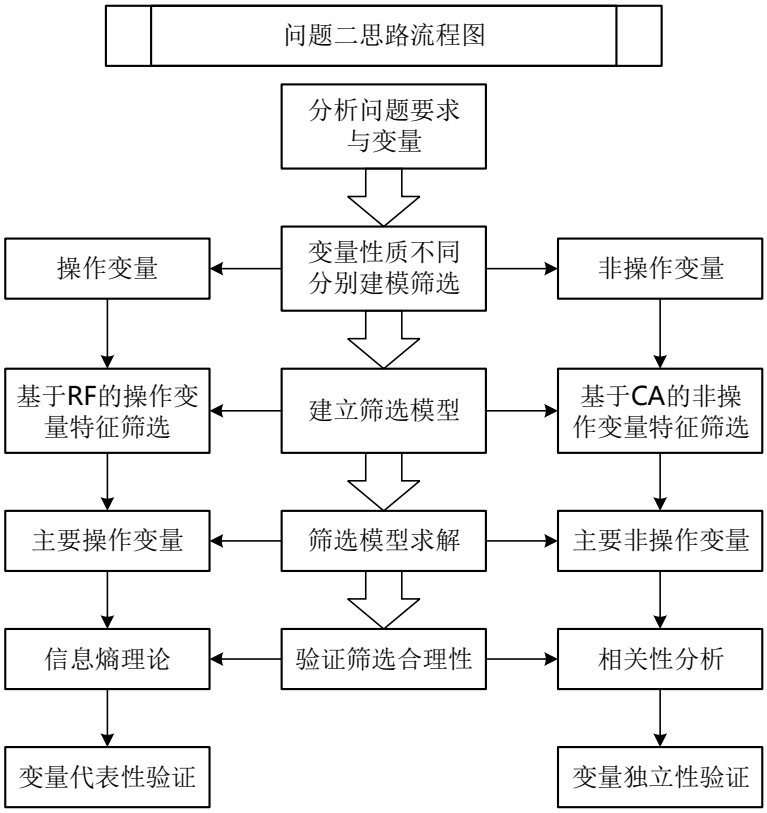

图 4-1 问题二思路流程图

4.2. 问题二模型建立

4.2.1. 基于随机森林的操作变量特征筛选模型

随机森林（Random Forest，简称 RF）是一种新兴起、高度灵活的机器学习算法，拥有广泛的应用前景，在大量分类以及回归问题中具有极好的准确率。并且，随机森林算法自带特征筛选机制，即随机森林能够评估各个特征在相应问题上的重要性。此处，辛烷值损失模型的操作变量具有数量多、高度非线性、相互强耦联的特点，与随机森林算法的应用条件契合，所以本文考虑建立基于随机森林的操作变量特征筛选模型，对预处理后的 350 个操作变量进行有效筛选。其具体步骤如下。

(1) 特征贡献度计算

基于随机森林的操作变量特征筛选模型中，特征重要性定量为每个特征在随机森林的每棵树上贡献度的平均值，通过比较不同特征之间的贡献度大小，来确定不同特征的重要程度。其中，贡献度用平均袋外数据误差表示：袋外数据误差如式(4-1)所示：

$$OobErrorT_i^j = |e_1 - e_2| \tag{4-1}$$

其中，$OobErrorT_i{}^j$ 为特征 j 对随机森林中树 i 的袋外数据误差，e_1 为用袋外数据样本得到的误差，e_2 为随机打乱袋外数据中的第 j 列后得到的误差。

利用式(4-1)计算平均袋外数据误差：

$$MOET^{j}=\frac{\sum_{i=1}^{T}OobErrorT_i}{T} \tag{4-2}$$

其中，$MOET^{j}$ 为特征 j 的平均袋外数据误差，T 为随机森林中树的数量。并且，涉及到的对袋外数据第 j 列进行随机打乱的方法采取以下模式：通过排列的方式将原来所有样本的第 j 个特征值重新打乱分布。

至此，可以用平均袋外数据误差来刻画特征 j 的重要性。其依据是：如果一个特征很重要，那么其波动会非常影响测试的误差，如果测试误差改变幅度较小，则说明特征 j 不重要。

(2) 随机森林训练

此处，基于随机森林的操作变量特征筛选模型中随机森林训练过程包含以下步骤：

Step1：原始训练集为 N，应用 bootstrap（有放回抽样）法有放回地随机抽取 k 个新的自助样本集，并由此构建 k 棵分类树，每次未被抽到的样本组成了 k 个袋外数据；

Step2：假如特征空间共有 D 个特征，则在每一轮生成决策树的过程中，从 D 个特征中随机选择 d 个特征（d<D）组成一个新的特征集，通过使用新的特征集来生成决策树，在 k 轮中共生成 k 个相互独立的决策树；

Step3：将生成的多棵树组成随机森林，相互独立的若干棵决策树的重要性是相等的，无需考虑它们的权值。

随机森林算法的流程图如图 4-2 所示。

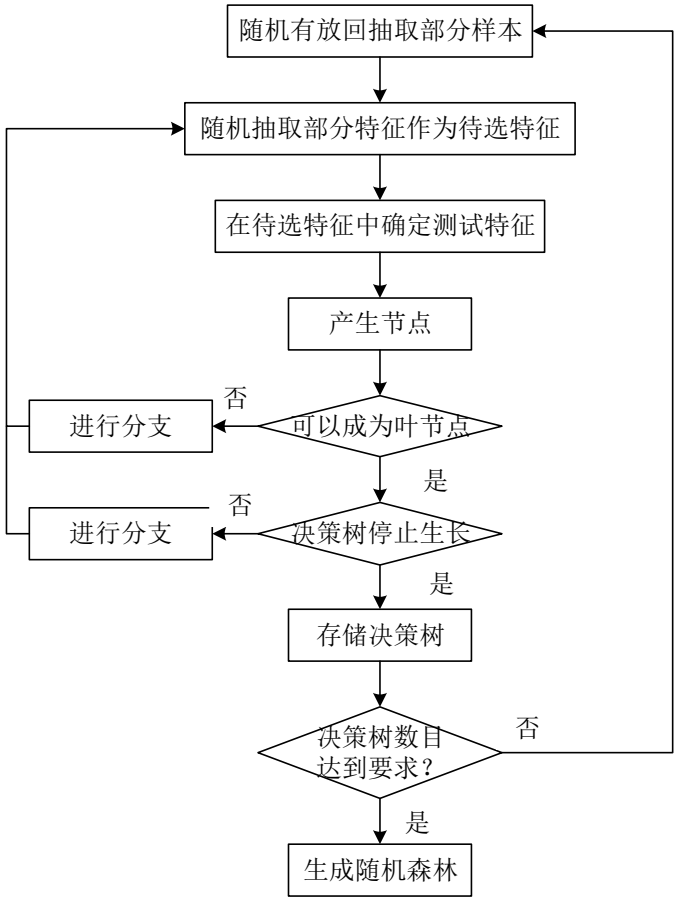

图 4-2 随机森林实现流程图

(3) 特征权值计算

特征权值反映了操作变量的重要程度占比，对每一个操作变量的特征权值，其表示为该特征的平均袋外数据误差与全体特征平均袋外数据误差之和的比值，具体数学表达式如下所示：

$$weight_j = \frac{MOET^j}{\sum_{k=1}^{D} MOET^k} \tag{4-3}$$

其中，$weight_j$ 为特征 j 的权值，$MOET^j$ 为特征 j 的平均袋外数据误差，D 为特征总数（此处为预处理后的操作变量总数 350）。

(4) 主要操作变量筛选

得到所有特征的特征权值后，通过对其进行降序排列得到操作变量特征权值集合 $WeightSet$（其中，$weight_{ranki}$ 为排名为 i 的变量的权值），即可根据需要筛选出前若干个对辛烷值损失贡献度较高的建模主要操作变量。

$$WeightSet = \begin{bmatrix} weight_{rank1} \\ weight_{rank2} \\ \vdots \\ weight_{rankD} \end{bmatrix} \tag{4-4}$$

4.2.2. 基于相关性分析的非操作变量特征筛选模型

相关性分析(Correlation Analysis，简称 CA)是一种研究数据集中属性与属性之间相关性的统计学方法。此处，辛烷值损失模型的非操作变量（原料性质变量、待生吸附剂性质变量、再生吸附剂性质变量）具有数量少，同类变量之间存在潜在化学相关性的特点，如果使用传统的降维（或筛选）方法对其进行操作则没有意义，所以本文考虑对原料性质变量和吸附剂性质（待生吸附剂性质、再生吸附剂性质）变量分别进行相关性分析，建立基于相关性分析的非操作变量特征筛选模型，保留相互间相关性较弱的变量，甄选相关性较强的变量，对预处理后的 7 个原料性质变量、2 个待生吸附剂性质变量、2 个再生吸附剂性质变量进行有效筛选。其具体步骤如下。

(1) 正态性检验

绘制附件 1 预处理后的 7 个原料性质变量、2 个待生吸附剂性质变量、2 个再生吸附剂性质变量的样本分布图，如图 4-3、图 4-4 所示：

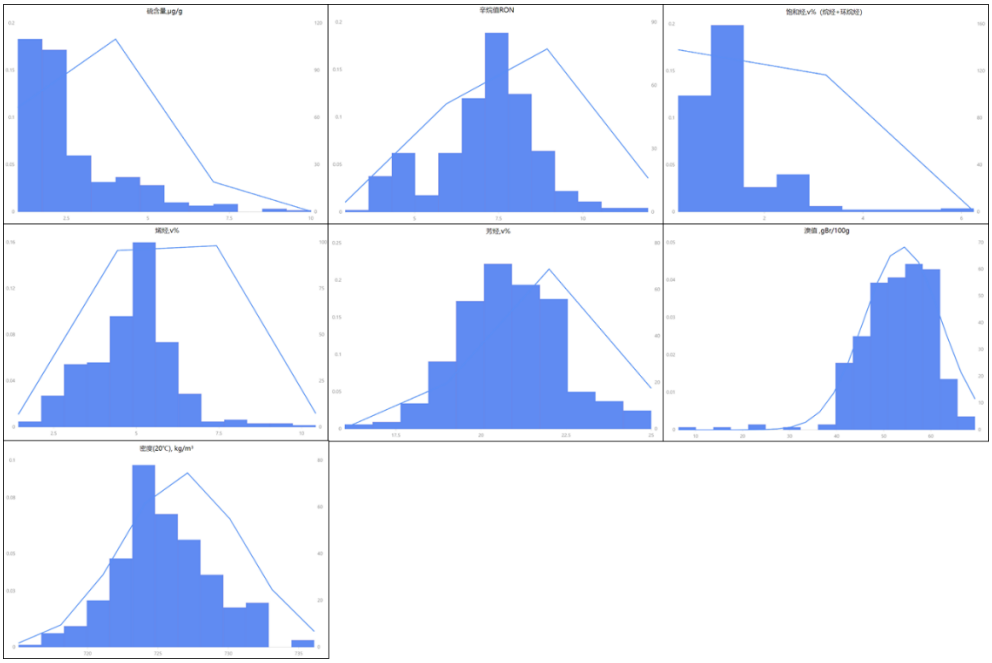

图 4-3 原料性质变量的样本分布图

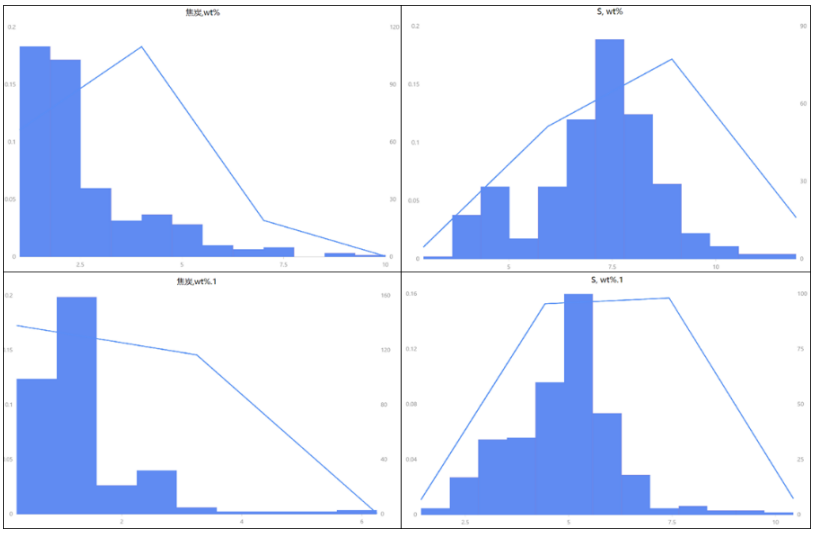

图 4-4 吸附剂性质变量的样本分布图

利用 Shapiro-Wilk 正态性检验方法对附件 1 预处理后的 7 个原料性质变量、2 个待生吸附剂性质变量、2 个再生吸附剂性质变量进行正态性检验，结果如表 4-1、表 4-2 所示（表中 0.000 表示在当前精度下近似为零）：

表 4-1 原料性质变量 Shapiro-Wilk 检验 p 值表

变量名	硫含量	辛烷值	饱和烃	烯烃	芳烃	溴值	密度(20℃)
P 值	0.002	0.000	0.000	0.000	0.043	0.000	0.011

表 4-2 吸附剂性质变量 Shapiro-Wilk 检验 p 值表

变量名	待生吸附剂性质		再生吸附剂性质	
	焦炭	S	焦炭	S
P 值	0.000	0.000	0.000	0.000

由上表可知，所有变量的 Shapiro-Wilk 正态性检验 p 值均小于 0.05，表明所有变量均不符合正态分布的假设，结合图 4-1、图 4-2 的样本分布图，亦可看出变量的样本分布不具有明显的正态性。

(2) Spearman 相关系数

由于所有变量的样本数据均不符合正态分布，所以考虑使用 Spearman 相关系数来衡量两个变量之间的相关性关系。斯皮尔曼(Spearman)相关系数对数据条件的要求没有其他相关系数严格，只要两个变量的观测值是可由连续变量观测值转化得到的等级值，不论两个变量的总体分布形态、样本容量的大小如何，都可以用 Spearman 相关系数来进行研究。

对于两个随机变量 X 和 Y，其样本数量均为 N，则变量 X 和 Y 之间的 Spearman 相关系数计算方法如下：

$$\rho=1-\frac{6\sum_{i=1}^{N}d_i^2}{N(N^2-1)} \tag{4-5}$$

其中，ρ表示两随机变量间的 Spearman 相关系数，d_i 表示两个变量分别排序后成第 i 个样本的等级差，N 表示样本总数。

(3) 主要非操作变量筛选

分别计算不同类型非操作变量（7 个原料性质变量、4 个吸附剂性质变量）两两之间的 Spearman 相关系数，得到变量 Spearman 相关系数矩阵：

$$Matrix_{\rho}=\begin{bmatrix}\rho_{11}\ \rho_{12}\cdots\rho_{1D}\\ \rho_{21}\ \rho_{22}\cdots\rho_{2D}\\ \vdots\\ \rho_{D1}\ \rho_{D2}\cdots\rho_{DD}\end{bmatrix} \tag{4-6}$$

其中，ρ_{ij}表示非操作变量 i 与非操作变量 j 之间的 Spearman 相关系数值，D 为当前类型非操作变量总数。

找出变量 Spearman 相关系数矩阵中所有 Spearman 相关系数值大于阈值α的位置，定义其行与列表示的非操作变量为强相关变量，结合实际分析并将其行和列代表的非操作变量中的一个剔除，即可筛选出若干个互不相关的建模主要非操作变量。

4.3. 问题二模型求解

4.3.1. 操作变量特征筛选模型求解

(1) 操作变量特征权值求解

利用 python 对上述基于随机森林的操作变量特征筛选模型进行求解，得到 350 个操作变量的特征权值（降序排列）如表 4-3 所示（完整表格见附表“问题 2-随机森林特征权值.xlsx”）：

表 4-3 操作变量特征权值表

变量编号	76	26	41	……	349	350
特征权值	0.2942	0.1397	0.0838	……	0.0000	0.0000

对上述 350 个操作变量的特征权值进行描述性统计分析，得到描述性统计分析表如表 4-4 所示：（上述和下述结果中 0.0000 均表示在当前精度下近似为零）

表 4-4 特征权值描述性统计表

统计量	最小值	最大值	平均值	标准差	中位数	众数
数值	0.0000	0.2942	0.0029	0.0190	0.0000	0.0000

作出 350 个操作变量的特征权值频数统计图，如图 4-5 所示：

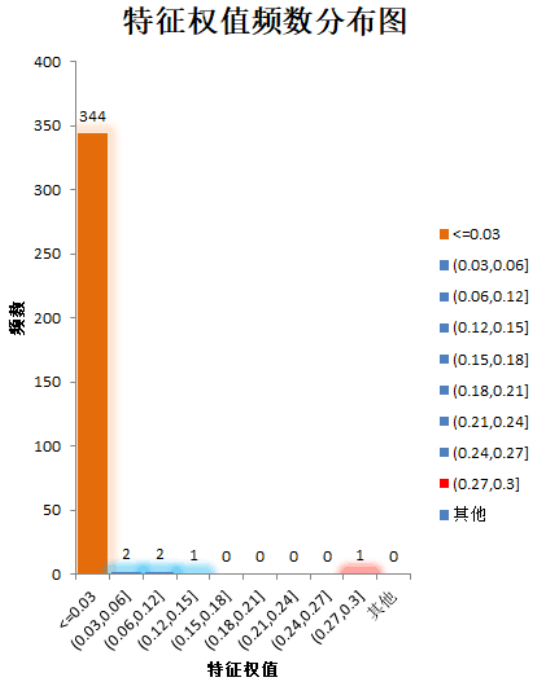

图 4-5 特征权值频数分布统计图

(2) 操作变量筛选结果

根据上述计算结果可知，全体操作变量中存在大量变量的特征权值为 0.0000（表示在当前精度下近似为零），这些变量对产品中辛烷值(RON)的贡献度几乎可以忽略不计，本文界定此类变量在误差允许的范围内为建模中的“无关变量”。其次，为了有效剔除次要变量，筛选主要变量，本文界定对产品中辛烷值(RON)的贡献度不大于 1%（特征权值不大于 0.01）的变量在误差允许的范围内为建模中的“次要变量”。以此，对操作变量中的“无关变量”和“次要变量”进行剔除，得到 350 个（原始为 354，经预处理后剩余 350）操作变量中的 16 个建模主要变量。如表 4-5 所示（变量按特征权值降序排列），详细变量中文名称、位号等信息见附表“问题 2-变量筛选结果”。

表 4-5 建模主要操作变量表

变量编号	76	26	41	295	274	245	287	269
特征权值	0.2942	0.1397	0.0838	0.0722	0.0499	0.0373	0.0157	0.0154
变量编号	70	39	64	132	253	339	20	343
特征权值	0.0245	0.0234	0.0212	0.0205	0.0195	0.0185	0.01	0.01

4.3.2. 非操作变量特征筛选模型求解

(1) 原料性质 Spearman 相关系数计算

按照 Spearman 相关系数计算方法计算原料性质 7 个变量两两之间的 Spearman 相关系数，结果如表 4-6 所示：

表 4-6 原料性质 Spearman 相关系数表

	硫含量	辛烷值	饱和烃	烯烃	芳烃	溴值	密度
硫含量	1.000	0.436	-0.450	0.413	-0.029	-0.169	0.134
辛烷值	0.436	1.000	-0.456	0.386	0.053	-0.047	0.222
饱和烃	-0.450	-0.456	1.000	-0.919	-0.002	0.090	-0.137
烯烃	0.413	0.386	-0.919	1.000	-0.354	0.030	-0.014
芳烃	-0.029	0.053	-0.002	-0.354	1.000	-0.265	0.341
溴值	-0.169	-0.047	0.090	0.030	-0.265	1.000	-0.232
密度	0.134	0.222	-0.137	-0.014	0.341	-0.232	1.000

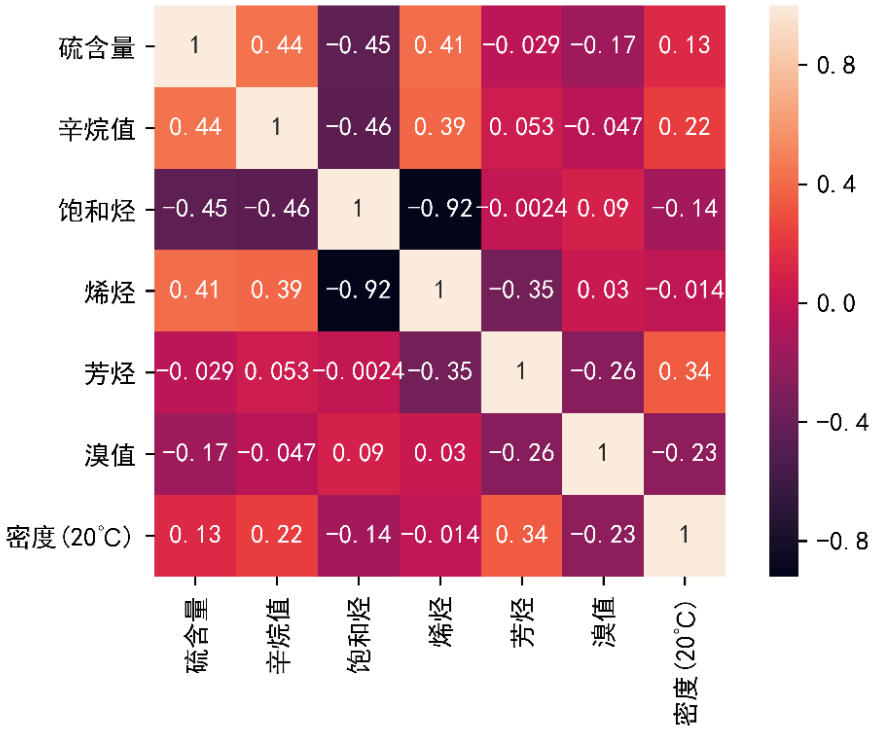

图 4-6 原料性质相关系数热力分析图

(2) 吸附剂性质 Spearman 相关系数计算

按照 Spearman 相关系数计算方法计算吸附剂性质 4 个变量两两之间的 Spearman 相关系数，结果如表 4-7 所示（其中焦炭-1、S-1 表示待生吸附剂性质，焦炭-2、S-2 表示再生吸附剂性质）：

表 4-7 吸附剂性质性质 Spearman 相关系数表

	焦炭-1	S-1	焦炭-2	S-2
焦炭-1	1.000	0.552	0.748	0.555
S-1	0.552	1.000	0.517	0.794
焦炭-2	0.748	0.517	1.000	0.664
S-2	0.555	0.794	0.664	1.000

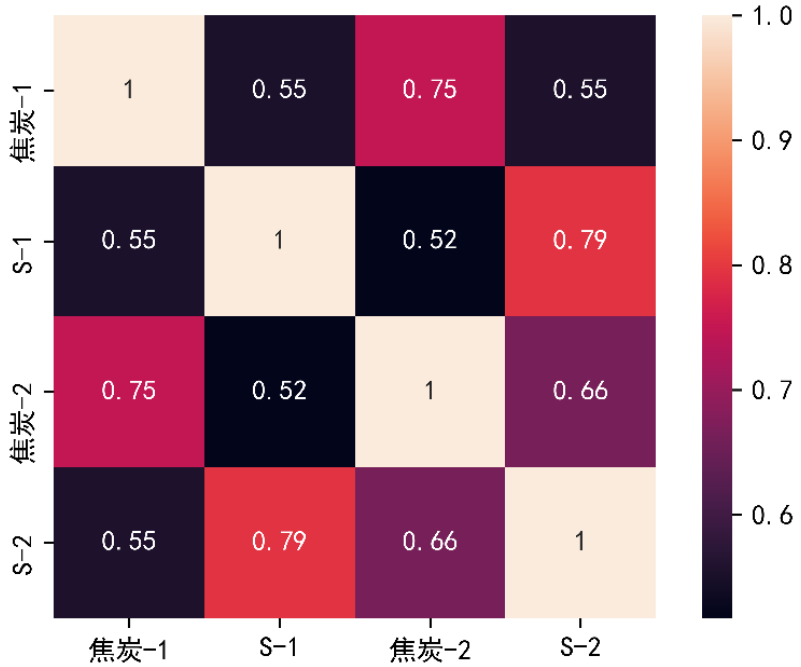

图 4-7 吸附剂性质相关系数热力分析图

（3）非操作变量筛选结果

由于非操作变量具有数量少，同类变量之间潜在化学相关性关系复杂的特点，不能轻易将其中变量进行剔除。所以，本文本着瓦尔德准则并结合 Spearman 相关系数对非操作变量（原料性质变量、吸附剂性质变量）进行筛选和剔除，即对于非操作变量“宁可放过勿轻删”。具体为：设置一个相对较高的阈值 0.8，对于 Spearman 相关系数大于 0.8 或小于-0.8 的两强相关变量进行单独分析，结合催化裂化汽油精制过程实际筛选出两者中对建模过程“贡献”更大的变量。

通过上述计算结果可知，非操作变量中仅存在原料性质中饱和烃（烷烃+环烷烃）和烯烃呈现强负相关$(\rho$=-0.919$)$关系，考虑在建模中应对其中之一予以剔除。单独对饱和烃（烷烃+环烷烃）样本和烯烃样本做描述性统计分析，得到表 4-8：

表 4-8 饱和烃和烯烃描述性统计分析表

名称/统计量	最小值	最大值	平均值	标准差	中位数	变异系数(CV)
饱和烃	43.240	63.400	52.676	4.584	53.250	0.087
烯烃	14.600	34.670	25.389	4.955	24.800	0.195

由上述图表可知，饱和烃和烯烃样本数据的极差和标准差都相差不大，但是饱和烃和烯烃样本的变异系数分别为 0.087 和 0.195，烯烃样本的变异系数较大（且大于 0.15），表示烯烃样本数据中可能存在不正常的情况。综合上述分析，考虑剔除原料性质变量中的“烯烃”，保留“饱和烃”。

4.3.3. 主要变量筛选结果

综合上述，本文对建模主要变量的筛选过程及结果总结如下：

（1）采用基于随机森林的操作变量特征筛选模型对问题一预处理后的 350 个操作变量进行筛选，共剔除 334 个操作变量，保留 16 个建模主要操作变量。

（2）采用基于相关性分析的非操作变量特征筛选模型对 11 个非操作变量进行筛选，剔除原料性质中的“烯烃”变量，保留 10 个建模主要非操作变量。

（3）总共筛选出 26 个建模主要变量。

（注：筛选出的 26 个建模主要变量详细信息见附表“问题 2-变量筛选结果”）

4.4. 特征筛选合理性验证

4.4.1. 信息熵理论—变量代表性分析

香农于 1948 年提出了“信息熵”理论，解决了对信息的量化度量问题，目前该理论已经在工程技术、社会经济等领域得到了非常广泛的应用。

一般来说，某个指标的信息熵越小，表明指标值的变异程度越大，相应提供的信息量越多，其在综合模型中所能起到的作用也越大；相反，某个指标的信息熵越大，表明指标值的变异程度越小，提供的信息量也越少，在综合模型中所起到的作用也越小。其中，对于某项指标的一组数据，其信息熵的计算方式如下所示：

$$E_j = -ln^{-1}(n)\sum_{i=1}^{n} p_{ij}\, lnp_{ij} \tag{4-7}$$

$$p_{ij} = Y_{ij} / \sum_{i=1}^{n} Y_{ij} \tag{4-8}$$

其中，E_j 为第 j 项指标的信息熵，n 为样本总数，Y_{ij} 为该指标的第 j 条样本数据。

本文考虑使用信息熵理论来验证主要变量筛选结果的代表性，具体步骤如下：

Step1：计算各个筛选后主要变量的信息熵；

Step2：计算各个筛选后主要变量的信息熵权重，其中信息熵权重的计算方式如下：

$$W_j = \frac{1-E_j}{n-\sum_{i=1}^{n} E_i}(j = 1,2,...,n) \tag{4-9}$$

其中，W_j 是第 j 个筛选后主要变量的信息熵权重，E_j 为第 j 项指标的信息熵，n 为样本总数。

Step3：计算各个筛选后主要变量的信息熵累积权重，并根据结果进行分析。

根据上述步骤，计算全部 361 个变量的信息熵，其中筛选后的主要变量的信息熵、信息熵权重和信息熵累积权重如表 4-9 所示（其中 Var i 代表第 i 个主要操作变量）：

表 4-9 信息熵相关信息统计表

名称	硫含量	辛烷值	饱和烃	芳烃	溴值	密度	焦炭-1
信息熵	0.9919	0.9995	0.9988	0.9989	0.9973	0.9995	0.9714
权重	0.0006	0.0000	0.0001	0.0001	0.0002	0.0000	0.0021
累积权重	0.0006	0.0006	0.0007	0.0008	0.0010	0.0011	0.0032
名称	S-1	焦炭-2	S-2	Var1	Var2	Var3	Var4
信息熵	0.9930	0.9573	0.9909	0.8047	0.8136	0.4647	0.9162
权重	0.0005	0.0032	0.0007	0.0146	0.0140	0.0401	0.0063
累积权重	0.0037	0.0069	0.0076	0.0223	0.0362	0.0764	0.0827
名称	Var5	Var6	Var7	Var8	Var9	Var10	Var11
信息熵	0.4399	0.7907	0.6546	0.7182	0.4438	0.9500	0.8066
权重	0.0420	0.0157	0.0259	0.0211	0.0417	0.0037	0.0145
累积权重	0.1247	0.1403	0.1662	0.1874	0.2291	0.2328	0.2473
名称	Var12	Var13	Var14	Var15	Var16		
信息熵	0.9899	0.7943	0.8511	0.8852	0.5356		
权重	0.0008	0.0154	0.0112	0.0086	0.0348		
累积权重	0.2481	0.2635	0.2747	0.2833	0.3181		

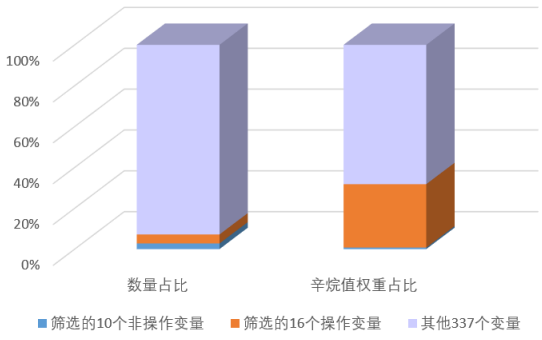

图 4-8 信息熵占比柱状图

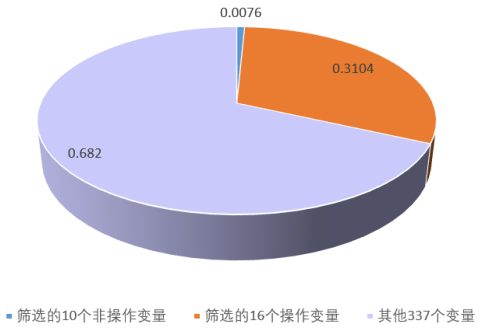

图 4-9 信息熵占比扇形图

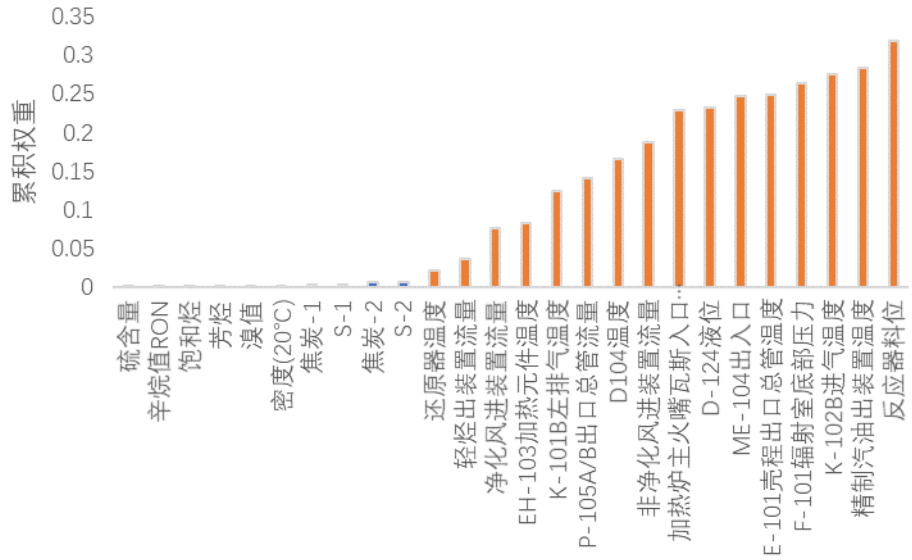

图 4-10 主要变量信息熵累积权重柱状图

由上述图表可知，26 个主要变量的累积信息熵权重为 0.3181，即 26 个主要变量的信息熵占比已达到全部 361 个变量信息熵总和的 31.81%（平均水平为 26/361=7.2%），由此，本文有理由说明所选择的 26 个主要变量具有一定的代表性。

4.4.2. 相关性分析—变量独立性分析

选择若干主要变量中的代表变量，绘制其两两之间的散点分布图，如图 4-11 所示。由图可知，变量与变量之间的散点分布并不存在某种特定的分布方式。

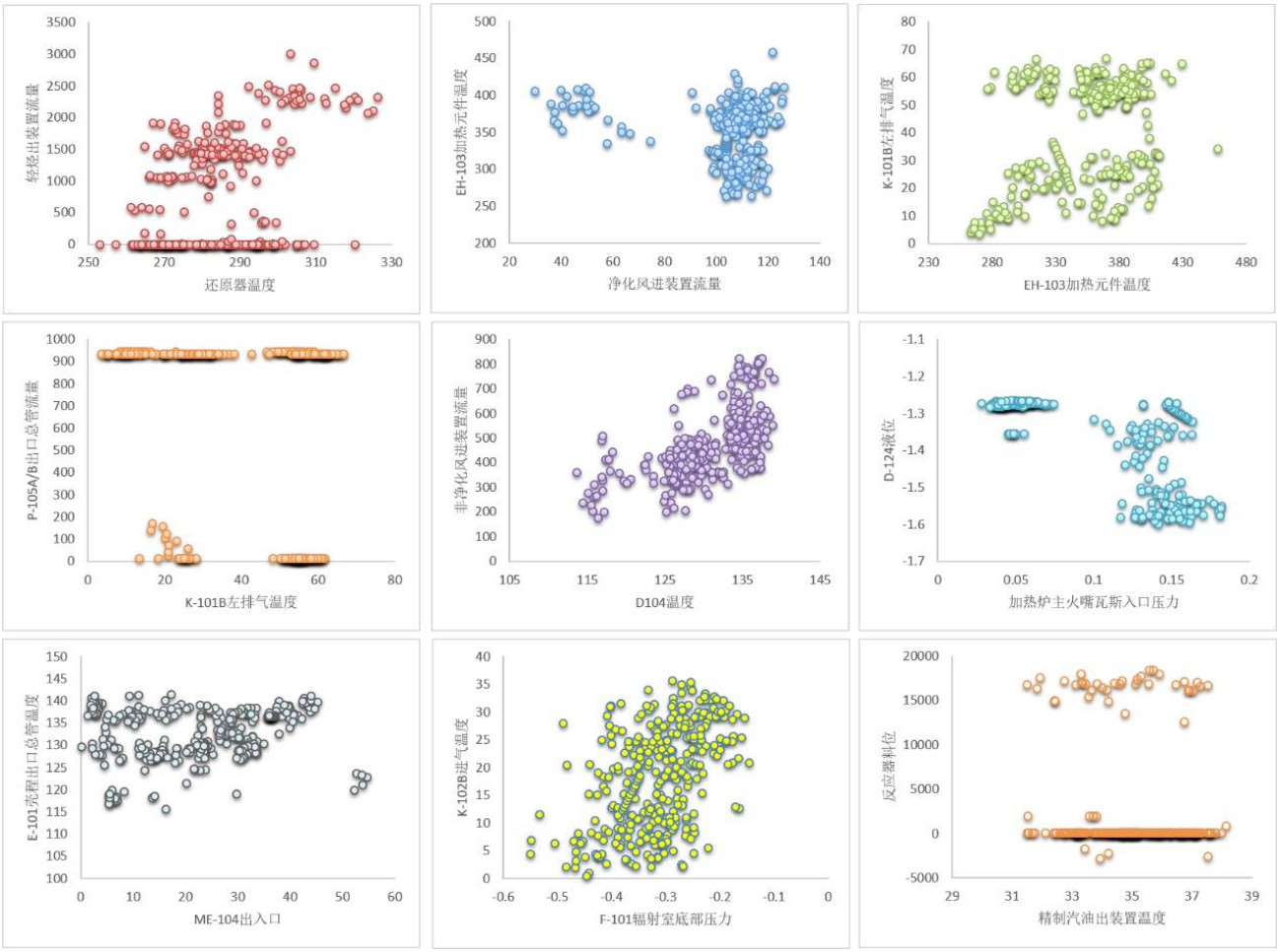

图 4-11 部分主要变量散点分布图

当主要变量两两之间相关性均较弱时，某种意义上任何主要变量均可独立的描述因变量的某方面性质，此时主要变量具有一定的独立性。参照 4.2.2 节中的方法计算 26 个建模主要变量两两之间的 Spearman 相关系数，得到其 Spearman 相关系数热力分析图如图 4-12 所示（其中 var *i* 代表第 *i* 个主要操作变量）：

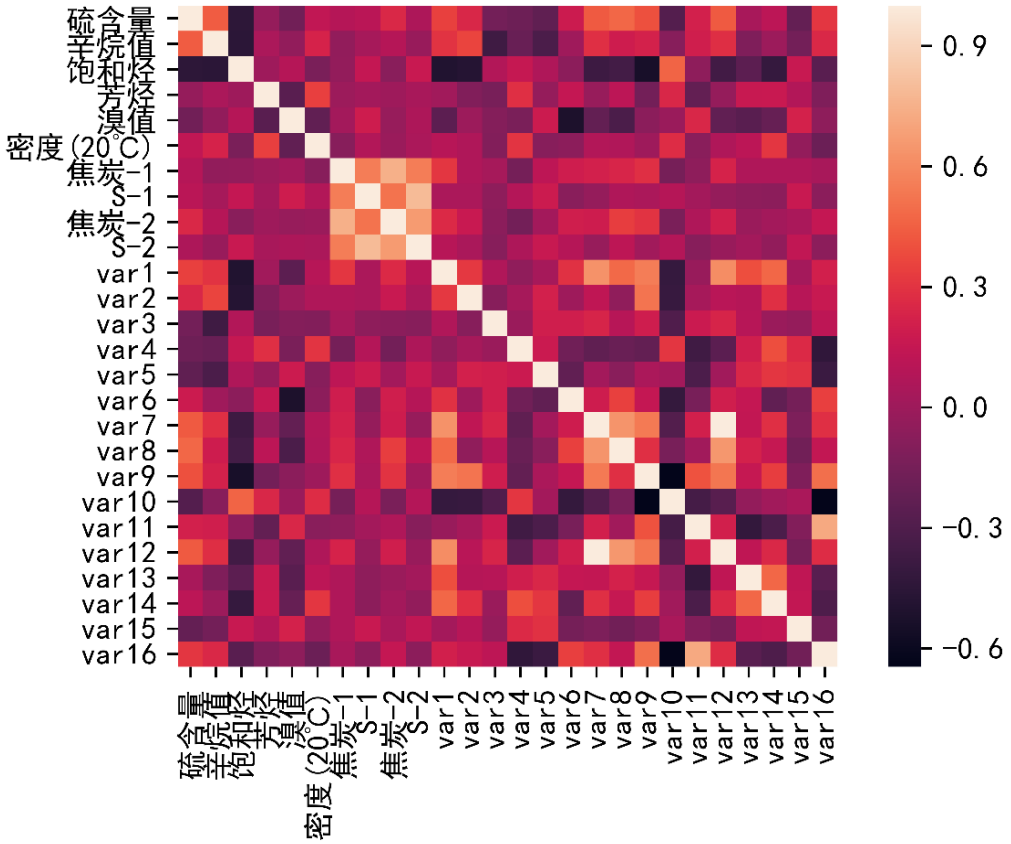

图 4-12 建模主要变量 Spearman 相关系数热力分析图

由上图可知，26 个建模主要变量两两之间均不存在较强相关性（图中少数吸附剂变量之间相关性较高已在 3.3.2 中予以说明），可认为这 26 个主要变量各自具有一定的代表性。

4.5. 结果结论

(1) 对辛烷值损失模型主要变量进行筛选，从 300 多个变量中共筛选出 26 个建模主要变量，其中包括 6 个原料性质变量、2 个待生吸附剂性质变量、2 个再生吸附剂性质变量和 16 个操作变量。

(2) 采用信息熵理论和相关分析对筛选后主要变量的代表性和独立性进行验证，阐明了筛选过程的合理性。

5. 问题三：辛烷值(RON)损失预测方法研究

5.1. 问题分析

针对问题三，要求建立辛烷值（RON）损失预测模型。考虑到辛烷值损失是由主要变量“原料辛烷值”经二次计算得到的，直接建模预测辛烷值损失则可能效果较差，所以考虑首先建立产品性质预测模型，再根据预测得到的产品辛烷值求出相应的辛烷值损失。其次，考虑到原料性质、吸附剂性质等主要变量可能存在潜在的化学联系机理，同时操作变量高度非线性、相互强耦联，此时传统的基于线性关系的回归预测模型可能不再适用；同样的，由于预测样本较少，一般的基于多层网络的机器学习回归预测模型也不适用。基于

上述考虑，本文采用一种基于浅层学习网络的回归预测模型——GBR(Gradient Boosting Regression)预测模型，以对产品性质进行有效预测。为验证上述分析并便于比较，本文也同时建立了多元线性回归模型和随机森林模型，通过三种模型的比较以验证 GBR 预测模型用于汽油精制过程预测的有效性与先进性。最后，进行模型的敏感性分析，探究主要变量与辛烷值损失之间的联系。问题三思路流程图如下。

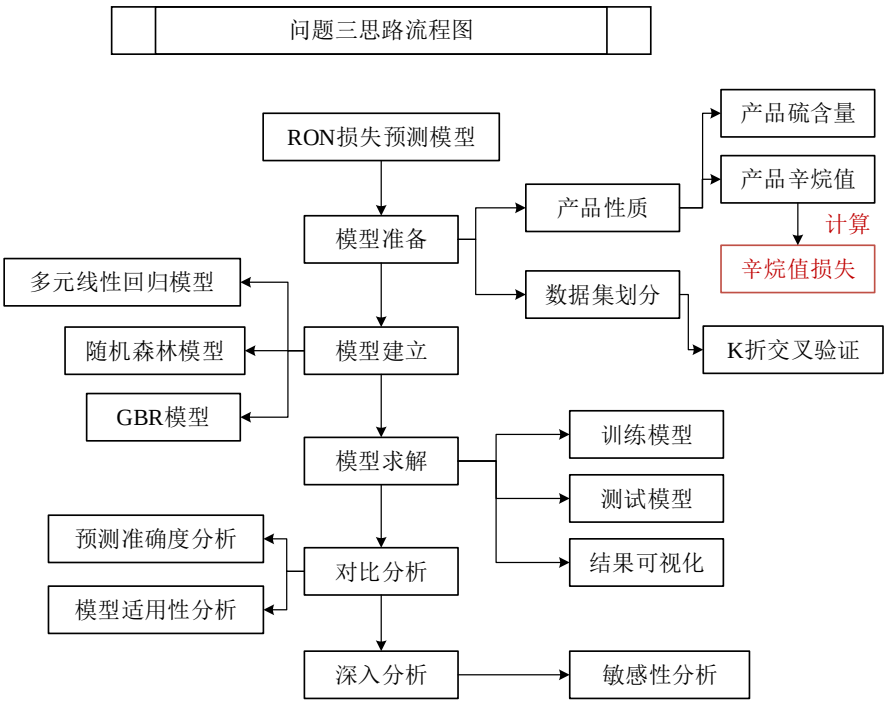

图 5-1 问题三思路流程图

5.2. 模型准备

5.2.1. 预测目标转换

根据题目分析，产品性质中辛烷值损失是由“原料辛烷值”与“产品辛烷值”经二次计算得到的（6.4.2 节中证明了其线性关系）。这种情况下，直接建模预测辛烷值损失值则会效果不佳，考虑对模型进行预测目标转换，即先建立产品辛烷值的预测模型，再根据产品辛烷值预测结果计算相应的辛烷值损失值。

5.2.2. 模型的交叉验证

在完成数据分析、预处理以及特征降维等步骤后，将进行模型的训练与验证。模型的训练与验证需要反复迭代进行。模型的训练与验证涉及到训练集、验证集以及测试集。训练集用于训练和调整模型参数；验证集用来检验模型精度和调整模型的超参数；测试集用来验证模型的泛化能力。由于本题中没有给定验证集，因此需要从训练集中拆分一部分作为验证集。验证集的划分包括留出法(Hold-Out)、交叉验证法(Cross Validation, CV)以及自助采样法(Bootstrap)。在进行模型验证时，通常采用的方法时留出法和交叉验证法，而留出法适用于数据量比较大的场景。本题中样本数据集仅有 300 多条，所以采用交叉验证法来实现模型的验证。

交叉验证法是将训练集划分为 K 份，将其中 K-1 份作为训练集，剩余 1 份作为验证集，

进行 K 次训练。这种划分方法是所有训练集都进行验证，最终模型验证精度是 K 份的平均值。交叉验证法的优点在于其验证集的精度比较可靠，训练 K 次可以得到 K 个有多样性差异的模型。

5.3. 问题三模型建立

5.3.1. 基于多元线性回归的产品辛烷值预测模型

回归分析(Regression Analysis)是一种统计学上分析数据的方法，目的在于了解两个或多个变量间是否相关、相关方向与强度，并建立数学模型以便观察特定变量来预测所需的变量。线性回归假设数据具有一定的线性关系，且线性关系越强，效果越好。

根据上述分析和相关数据表明，多种操作变量均会对产品的辛烷值产生影响。为了能够根据这些操作变量对产品的辛烷值进行预测，本文首先将尝试构建基于多元线性回归的产品辛烷值预测模型。多元线性回归模型构建过程如下：

样本数据集 $x^{(i)}=\left(X_1^{(i)},X_2^{(i)},\dots X_n^{(i)}\right)$ 对应的是一个向量，每一行是一个样本，每列对应一个特征。对应的结果可以用如下公式表示：

$$y=\theta_0+\theta_1x_1+\theta_2x_2+\cdots+\theta_nx_n \tag{5-1}$$

模型的求解思路与简单线性回归一致，即已知训练数据样本，找到对应的参数 $\theta=\left(\theta_0,\theta_1,\theta_2,\cdots,\theta_n\right)$，该向量是列向量，可以虚构第 0 个特征 x_0，令其恒等于 1，则推导时结构更加整齐：

$$y=\theta_0X_0^{(i)}+\theta_1X_1^{(i)}+\theta_2X_2^{(i)}+\cdots+\theta_nX_n^{(i)} \tag{5-2}$$

将上式改写成向量点乘的形式：

$$X=\begin{pmatrix}1 & X_1^{(1)} & X_2^{(1)} & \cdots & X_n^{(1)}\\ 1 & X_1^{(2)} & X_2^{(2)} & \cdots & X_n^{(2)}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & X_1^{(n)} & X_2^{(n)} & \cdots & X_n^{(n)}\end{pmatrix}\quad \theta=\begin{pmatrix}\theta_0\\ \theta_1\\ \theta_2\\ \vdots\\ \theta_n\end{pmatrix} \tag{5-3}$$

因此，我们可以把目标写成向量化的形式：

已知训练数据样本 x,y,找到向量 θ，使 $\left(y-X\bullet\theta\right)^T\left(X\bullet\theta\right)$ 尽可能小。推导可以得到多元线性回归的正规方程解：

$$\theta=\left(X^TX\right)^{-1}X^Ty \tag{5-4}$$

由此可得到基于多元线性回归的产品辛烷值预测模型。利用该模型，即可求得相应变量条件下的产品辛烷值。

5.3.2. 基于随机森林的产品辛烷值预测模型

随机森林算法是一种重要的集成学习方法。随机森林使用多个 CART 决策树作为弱学

习器，不同决策树之间没有关联。进行分类任务时，森林中的每一棵决策树分别对输入样本进行判断和分类，每棵决策树都会得到一个分类结果，随机森林算法将选择被分类次数最多的类别当作最终的分类结果。

随机森林在生成决策树时用随机选择的特征，即使用 bagging 方法。因为如果训练集中的某几个特征对输出的结果有很强的预测性，那么这些特征会被每个决策树所应用，导致树之间具有相关性，不会减小模型的方差。随机森林的建立过程如下：

Step1：原始训练集中有 N 个样本，且每个样本有 W 维特征。从数据集中有放回的随机抽取 x 个样本组成训练子集，一共进行 w 次采样，即生成 w 个训练子集；

Step2：每个训练子集形成一棵决策树，一共形成 w 棵决策树；

Step3：对于单个决策树，树的每个节点处从 M 个特征中随机挑选 m 个特征，按照结点不纯度最小原则进行分裂。每棵树均按照此方式分裂下去，直到该节点的所有训练样本均属于同一类。在决策树的分裂过程中不需要剪枝；

Step4：根据生成的多个决策树分类器对需要进行预测的数据进行预测。对于回归树来说，根据每棵树的投票结果，利用简单的平均值得到最终结果。

5.3.3. 基于 GBR 的产品辛烷值预测模型

梯度提升回归（Gradient boosting regression，GBR）是一种集成式的回归算法，它由多种弱学习器组成[1]。梯度提升回归基于 Gradient boosting（GB）框架，其思想是将已有的回归算法依次集成，得到一个性能强大的算法。由于其突出的预测能力，已被广泛用于处理非线性问题[2]；有研究表明，对于样本量较少、变量多变量之间强耦合的场景，梯度提升回归模型比支持向量回归、随机森林和多层感知器模型提供了更好的预测精度[3]。
梯度提升回归算法每次迭代会产生一个新模型，使用损失函数来评估模型对训练集的准确程度，模型应尽可能让损失函数最小[4]。这里，我们使用均方根误差作为损失函数：

$$Loss = MSE = \sum (y_i - y_i^p)^2 \tag{5-5}$$

为了减小损失函数，算法采用梯度下降的方法，基于残差进行学习，每次迭代朝着损失函数的负梯度方向移动，来找到使得 MSE 最小的值。不断迭代训练直到 MES 足够小。每层迭代时，视其对训练集中每个样本的预测结果、训练集总体预测准确率来修正每个样本的权值，下一层学习器的训练将使用上一层修正过权值的数据集。为了获得更快的迭代效果，我们对学习器制定了函数递减的“贪婪阶段”方法，在每次迭代时指定最佳步长ρ。对于第 t 次迭代时的函数估计，优化规则定义为：

$$f_t \leftarrow f_{t+1}\rho_t h(\mathrm{x}, \theta_t) \tag{5-6}$$

$$(\rho_t, \theta_t) = \mathrm{argmin} \sum_{i=1}^{N} \psi(y_i, f_{t-1}) + \rho h(\mathrm{x}_i, \theta) \tag{5-7}$$

最后，将每层学习器按照分配的权值融合在一起，得到最终的模型。

梯度提升回归集成的每个算法的准确率都不高，通过建立多层优化模型，针对同一个训练集每层训练一个弱学习器，其中每层计算的目的是减少上一层计算的残差，在残差减少的梯度方向上创建一个新的模型，以获得比单一学习器更优越的泛化性能。

5.4. 问题三模型求解

5.4.1. 基于多元线性回归的产品辛烷值预测模型求解

利用 Python 求解基于多元线性回归的产品辛烷值预测模型，首先建立散点图查看不同主要变量与预测变量之间的线性相关程度。分别选择原料辛烷值 RON、S-ZORB.TC_2801.PV、S-ZORB.PDT_1004.DACA 作为自变量，产品辛烷值 RON 作为因变量，线性情况如图 5-2 所示，并给出最佳拟合直线和 95%的置信带。

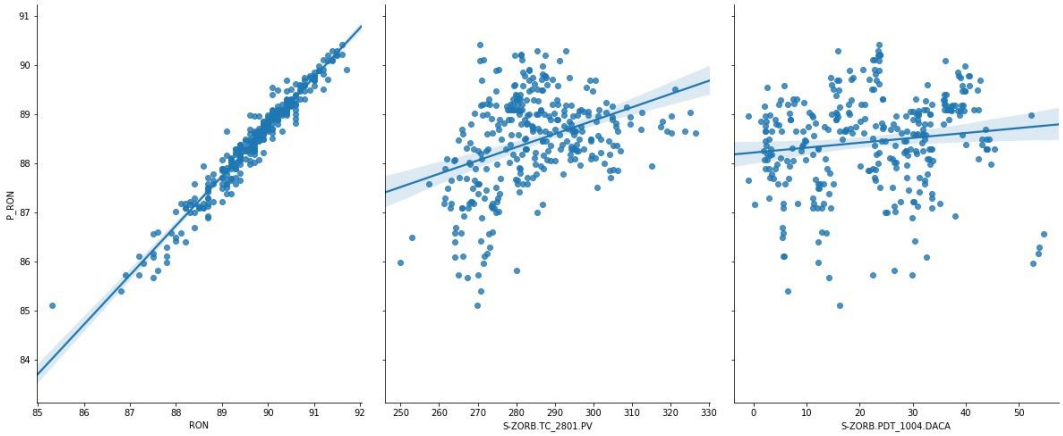

图 5-2 不同主要变量对应产品辛烷值的线性情况示意图

从图中可以看出，存在部分变量与产品辛烷值有比较能够强的线性关系，而另外一些变量线性关系较弱，但也属于强相关。此外，我们还可以了解到不同的主要变量对产品辛烷值的预测趋势（置信度=95%），接下来将采用 26 个主要变量以及产品辛烷值建立多元线性回归方程，计算各主要变量对应的回归系数，得到回归方程如下：

$$y = 0.34610 + 9.86353e\text{-}05x_1, +0.936629x_2 - 0.00158681x_3 + 0.0061287x_4 + 0.000828369x_5 + 0.00482861x_6 + 0.0246149x_7 - 0.0149961x_8$$
$$-0.0583471x_9 + 0.0293732x_{10} + 0.0051294x_{11} + 4.20617e\text{-}05x_{12} + 0.000285584x_{13} + 0.000247016x_{14} - 0.00320154x_{15} - 8.80513e\text{-}05x_{16} - 0.0312086x_{17}$$
$$+0.000135891x_{18} + 1.55492x_{19} + 0.370969x_{20} - 0.0034605x_{21} + 0.0323758x_{22} + 0.288236x_{23} - 0.00410545x_{24} - 0.0192308x_{251} + 4.93774e\text{-}07x_{26}$$

采用上述回归方程对数据集进行预测，预测结果如图 5-3 所示。

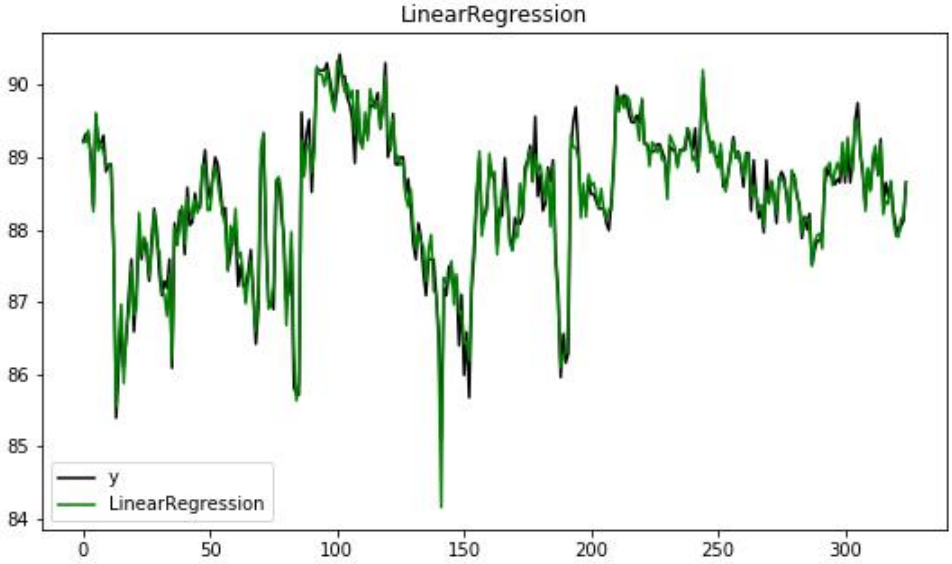

图 5-3 基于多元线性回归的产品辛烷值预测结果图

5.4.2. 基于随机森林的产品辛烷值预测模型求解

利用 Python 求解基于随机森林的产品辛烷值预测模型，预测结果如图 5-4 所示。

图 5-4 基于随机森林的产品辛烷值预测结果图

5.4.3. 基于梯度提升回归法的产品辛烷值预测模型求解

利用 Python 求解基于 GBR 的产品辛烷值预测模型，并保存模型以便后续建模使用。预测结果如图 5-5 所示。

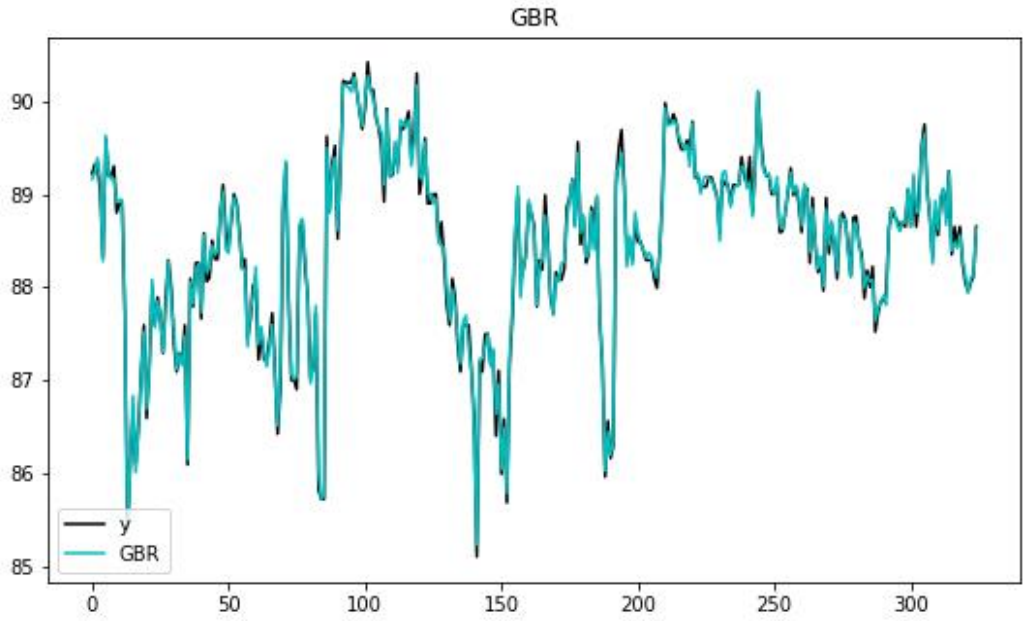

图 5-4 基于 GBR 的产品辛烷值预测结果图

5.5. 模型验证与结果分析

采用交叉验证对上述各预测模型进行验证。由于本文中样本数量较少，因此采用 5 折

交叉验证。三种模型各次交叉验证得分如表 5-1 所示。由表可知，三种预测模型均具有出良好的预测效果。从图 5-5 中可以看出，绝大多数情况下，GBR 模型的得分高于多元线性回归以及随机森林。

表 5-1 各预测模型交叉验证得分表

预测模型	得分 1	得分 2	得分 3	得分 4	得分 5
多元线性回归	0.9399	0.9670	0.8920	0.8462	0.6107
随机森林	0.9327	0.9475	0.8838	0.8856	0.7723
GBR	0.9413	0.9473	0.8854	0.7045	0.8273

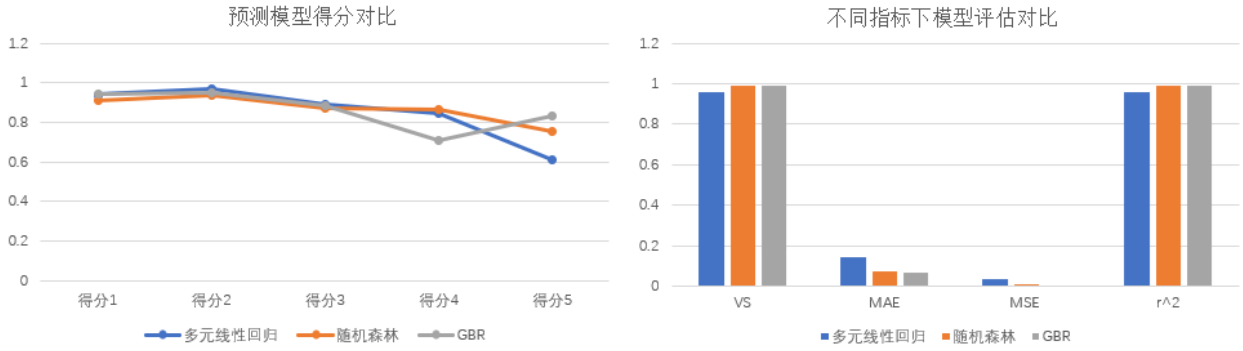

图 5-5 各预测模型评估对比

选择回归模型的方差得分(VS)、平均绝对误差(MAE)、均方差(MSE)以及判定系数四类指标对预测模型进行评估。

其中，模型的方差得分取值范围为[0,1]，越接近 1 说明自变量越能解释因变量的方差变化，其值越小说明效果越差；平均绝对误差用于评估预测结果和真实数据集的接近程度，其值越小说明拟合效果越好；均方差是表示拟合数据与原始数据对应样本点的误差的平方和的均值，其值越小说明拟合效果越好；判定系数的含义是解释回归模型的方差得分，其值越小说明效果越差。

因此，从表 5-2 和图 5-5 中可知，GBR 预测模型的拟合效果最优，其次为随机森林，效果最差的是多元线性回归。

表 5-2 各预测模型评估表

预测模型	回归模型 方差得分(VS)	平均绝对 误差(MAE)	均方差 (MSE)	判定系数(r^2)
多元线性回归	0.9617	0.1445	0.03698	0.9617
随机森林	0.9882	0.0748	0.0115	0.9881
GBR	0.9920	0.0692	0.0076	0.9920

5.6 结果结论

(1) 分别基于多元线性回归、随机森林、GBR 构建产品辛烷值预测模型。其中，基于 GBR 的产品辛烷值预测模型表现出最优的预测效果。

(2) 根据样本数量，采用 5 折交叉验证方法对所构建的产品辛烷值预测模型进行验证。从验证结果可以得到，各预测模型均表现出较好的泛化性，且预测稳定性较高。

5.7 敏感性分析

为探究建模主要变量与产品辛烷值（RON）损失之间的内在联系（5.2 节中证明产品辛烷值与辛烷值损失存在简单的线性关系，所以本节中考虑以数值分布更广的产品辛烷值作为研究因变量），对模型进行敏感性分析。

以 133 号样本为例，采用控制变量法保持其他变量取值不变，将 26 个建模主要变量在-50%至+50%幅度范围内以 5%为步长进行波动，每次波动后通过本章建立的预测模型预测产品辛烷值大小，得到产品辛烷值与变量波动幅度之间的结果表和敏感性分析图（完整波动范围数值结果见附表“问题 3-敏感性分析.xlsx”），如下所示：（其中 var *i* 代表第 *i* 个主要变量）

表 5-3 敏感性分析结果表

变量序号/波动幅度	-50%	-30%	-10%	10%	30%	50%
var1	87.91	87.91	87.98	87.98	87.98	87.98
var2	85.79	85.79	85.79	90.01	90.01	90.01
var 3	88.00	88.00	88.00	87.89	87.74	87.74
var 4	87.94	87.94	87.98	87.98	87.98	87.98
var 5	87.99	87.99	87.99	87.98	87.98	88.01
var 6	87.98	87.98	87.98	87.98	87.98	87.98
var 7	87.99	87.97	87.99	87.98	87.99	87.99
var 8	87.97	87.98	87.98	87.98	87.98	87.98
var 9	87.98	87.98	87.98	87.97	87.93	87.93
var 10	88.01	87.98	87.98	87.97	88.01	88.01
var 11	87.98	87.98	87.98	88.04	88.07	88.07
var 12	87.98	87.98	87.98	87.98	87.98	87.98
var 13	88.00	88.00	88.00	87.98	87.98	87.98
var 14	87.97	87.97	87.98	87.97	87.98	87.98
var 15	88.00	87.98	87.98	87.96	87.91	87.91
var 16	87.98	87.98	87.98	87.98	87.98	87.98
var 17	87.94	87.94	87.94	87.99	88.18	88.18
var 18	87.98	87.98	87.98	87.98	87.98	87.98
var 19	87.97	87.97	87.98	88.09	88.10	88.13
var 20	87.94	87.94	87.94	87.98	87.98	87.98
var 21	87.98	87.98	87.98	87.98	87.98	87.98
var 22	87.88	87.88	87.88	88.04	88.05	88.05
var 23	87.97	87.97	87.98	87.98	87.98	87.98
var 24	87.98	87.98	87.98	87.98	87.98	87.98
var 25	87.98	87.98	87.98	87.97	87.97	87.97
var 26	88.01	88.01	88.01	87.98	87.98	87.98

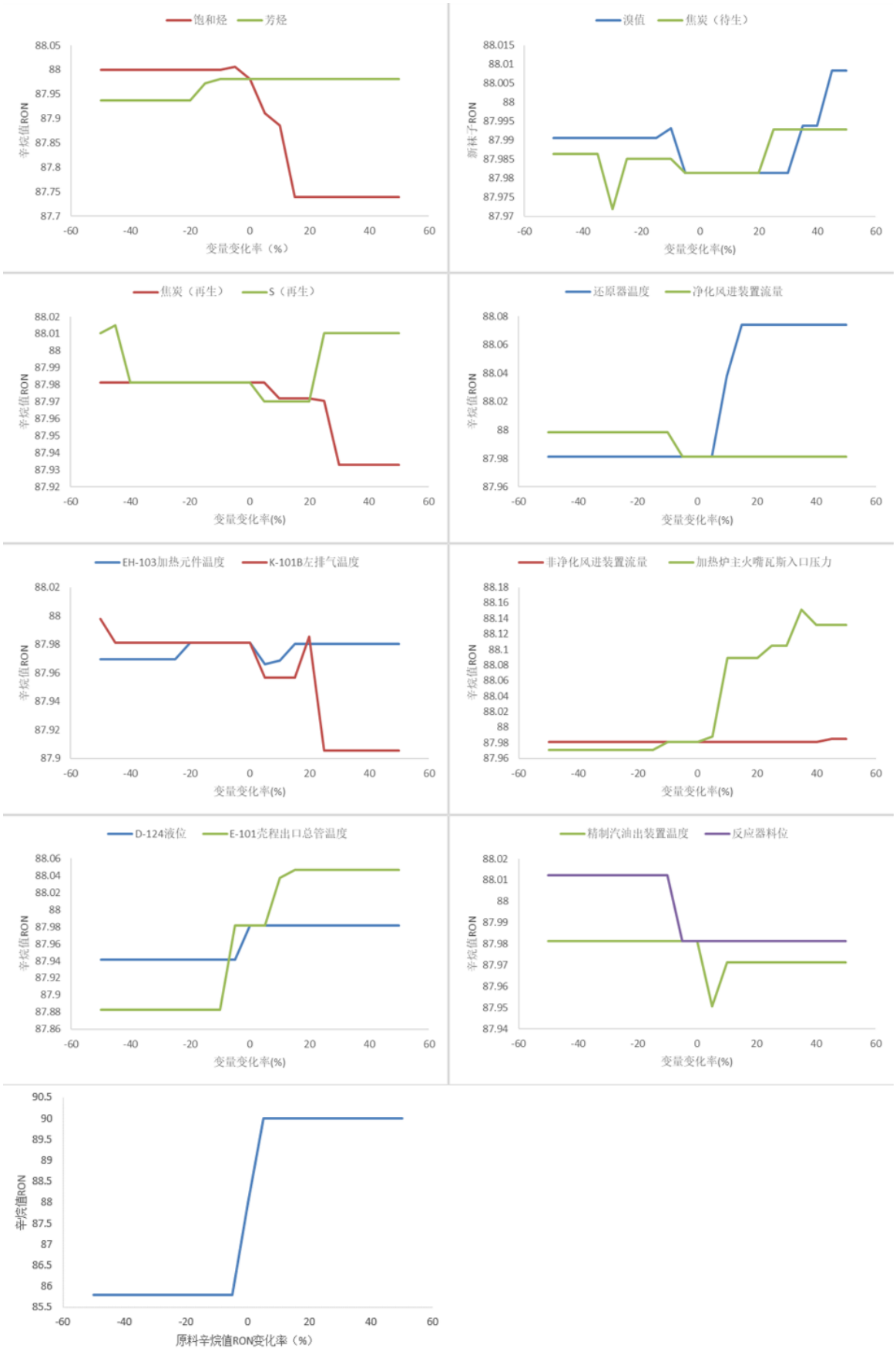

图 5-6 产品辛烷值与变量波动幅度敏感性分析图

由上述图表可知，在误差允许的范围内：
（1）在部分主要变量（如芳烃、净化风进装置流量等）不同幅的波动下，产品辛烷值（RON）变化幅度不大。
（2）在部分主要变量（如原料辛烷值、饱和烃、k-1018 左排气温度等）不同幅度的波动下，产品辛烷值（RON）变化剧烈。
（3）在部分主要变量（如 D-124 液位、还原器温度等）不同幅度的波动下，产品辛烷值（RON）呈现规律性变化。

6. 问题四：主要变量操作方案优化方法研究

6.1. 问题分析

针对问题四，要求对主要变量操作方案进行优化。考虑以最大化辛烷值（RON）损失降幅为目标函数，以问题二筛选后的 16 个操作变量为决策变量，添加产品硫含量约束以及决策变量取值范围约束，建立主要变量操作方案优化的线性规划模型。模型求解方面，考虑采用一种广泛用于优化问题的自适应和声搜索算法对该问题进行求解，并针对算法中的和声编码方式、和声解码方式和相关参数自适应调整方式进行设计，使得算法与问题的契合度更高。然后，对 325 个样本进行求解，对结果中辛烷值（RON）损失降幅大于 30%的样本单独进行分析，探究主要变量优化后的操作条件及相关规律。最后，为了验证自适应和声搜索算法的有效性与优越性，将其与一种基于二分法的启发式算法进行对比。问题四思路流程图如下：

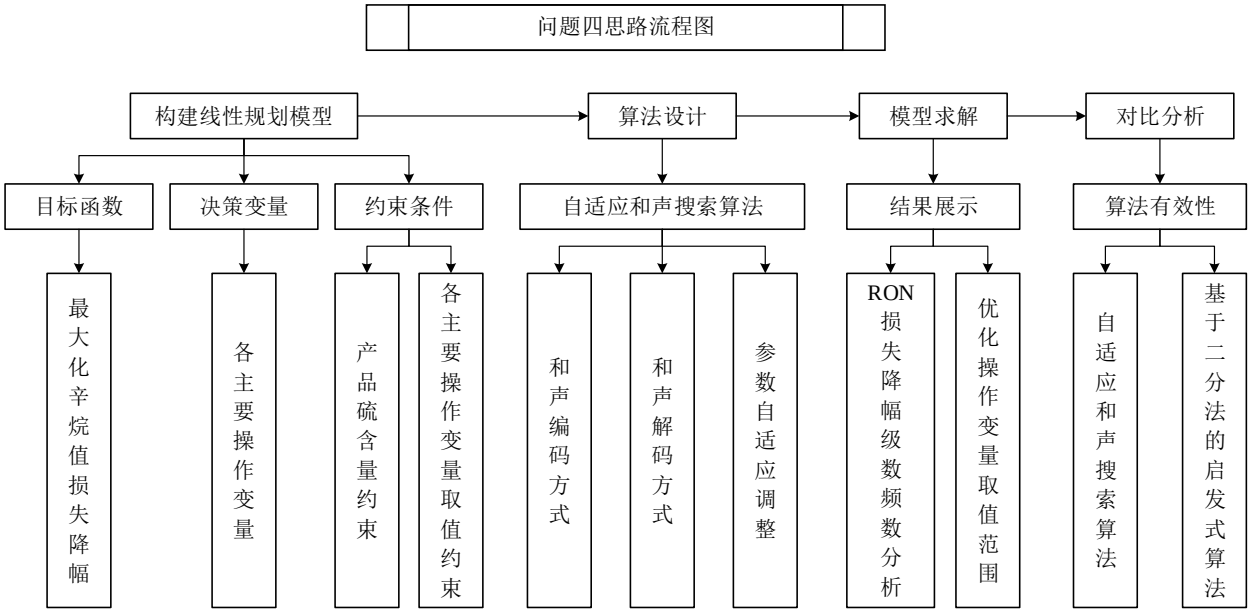

图 6-1 问题四思路流程图

6.2. 模型准备

根据题目要求，在针对主要变量进行优化的同时需要保证产品硫含量不大于 5 $\mu g / g$。而硫与辛烷值相同，均属汽油精制过程中的产品，需要根据当前操作的主要变量取值进行回归预测。所以同问题三，建立基于 GBR 的硫含量预测模型，以便于后续建模。

6.3. 问题四模型建立

(1) 目标函数

针对该问题，确定线性规划模型的目标函数为最大化辛烷值（RON）损失降幅，其表达式如下所示：

$$\max Z = \left| \frac{\Delta RON_{before} - \Delta RON_{after}}{\Delta RON_{before}} \right| \tag{6-1}$$

其中，ΔRON_{before} 表示优化操作变量之前的辛烷值损失，ΔRON_{after} 表示优化操作变量之后的辛烷值损失。优化操作变量前后的辛烷值损失计算公式如下所示：

$$\Delta RON_{before} = RON - RON_{before} \tag{6-2}$$

$$\Delta RON_{after} = RON - RON_{after} \tag{6-3}$$

其中，RON 表示原料辛烷值，RON_{before} 表示优化操作变量之前的产品辛烷值，RON_{after} 表示优化操作变量之后的产品辛烷值。RON_{after} 的计算公式如下所示：

$$RON_{after} = f_{RON}(x_1, x_2, \cdots x_{16}) \tag{6-4}$$

其中，$x_i\,(i = 1, 2, \ldots 16)$ 表示 16 个主要操作变量，f_{RON} 表示基于 GBR 的产品辛烷值预测模型。

(2) 决策变量

该线性规划模型的决策变量为 16 个主要操作变量 $x_i\,(i = 1, 2, \ldots 16)$，其每一组取值表示相应一种主要变量操作方案。

(3) 产品硫含量约束

根据题目要求，为了给企业装置操作留有空间，要求产品的硫含量不大于 5 μg/g，构建关于产品硫含量的约束条件如式(6-5)所示：

$$S_{after} \le 5 \tag{6-5}$$

其中，S_{after} 表示优化操作变量之后的产品硫含量，优化操作变量之后产品硫含量的计算公式如下所示，f_s 表示基于 GBR 的产品硫含量预测模型：

$$S_{after} = f_s(x_1, x_2, \cdots x_{16}) \tag{6-6}$$

(4) 各主要操作变量取值约束

根据附件 4 中所给出的各操作变量的取值范围信息，构建各主要操作变量的取值约束，如式(6-7)所示：

$$x_{i\min} \le x_i \le x_{i\max} \tag{6-7}$$

其中，$x_{i\min}$ 表示各操作变量可取到的最小值，$x_{i\max}$ 表示各操作变量可取到的最大值。

(5) 综合模型

综上所述，针对问题四我们建立了如下的线性规划模型：

$$\max Z = \left| \frac{\Delta RON_{before} - \Delta RON_{after}}{\Delta RON_{before}} \right|$$

$$s.t. \begin{cases} S_{after} \le 5 \\ x_{i\min} \le x_i \le x_{i\max} \\ S_{after} = f_s(x_1, x_2, \cdots x_{16}) \\ \Delta RON_{before} = RON - RON_{before} \\ \Delta RON_{after} = RON - RON_{after} \\ RON_{after} = f_{RON}(x_1, x_2, \cdots x_{16}) \\ i = 1, 2, \ldots 16 \end{cases}$$

6.4. 问题四模型求解

针对上述线性规划模型，采用和声搜索算法进行求解。和声搜索算法（HS）是 Geem 等人受音乐创作过程的启发，提出的一种元启发式全局搜索算法。音乐的创作过程源于不同乐器的不同音调组合成的和声的不同尝试，而为了寻找完美的和声，音乐家需要不断将调整各个乐器的音调，以求找到音调的最佳组合。同样，优化算法也是不断调整决策变量的组合以求到达最优目标值，与音乐创作过程有异曲同工之处。

6.4.1. 和声搜索算法介绍

(1) 和声搜索算法原理

在最优和声搜寻过程中，新和声中各个音调有三种产生途径：在已有和声对应音调中随机选取一个；在所有可能的音调中随机选择一个；对新产生的音调进行微调。如果新产生的和声，在乐理评价上比已经记录的所有和声中效果最差的和声好，则用该和声替换记录中评价最差的和声，否则舍弃新和声，直接进行下一轮的创作。如此重复练习直至创作出满足要求的和声或者达到最大创作次数为止，此时已经记录的所有和声中效果最好的和声则为本次创作中的最佳和声。表 6-1 给出音乐创作过程与 HS 算法寻优过程的类比。

表 6-1 音乐创作与算法寻优类比表

音乐创作过程	HS 算法寻优过程
音调	决策变量
和声	解向量
乐理评价	目标函数
练习过程	迭代过程

(2) 和声搜索算法相关概念

和声记忆库：和声记忆库（Harmony Memory, HM）是用于存放和声的集合。

和声记忆库大小：和声记忆库大小（Harmony Memory Size, HMS）是和声记忆库中存储和声的数目。其反应和声搜索算法的记忆能力。

记忆库选择概率：记忆库选择概率（Harmony Memory Considering Rate, HMCR）是新和声产生过程中，新和声对应音调从和声记忆库中对应位置进行随机选取的概率。

微调概率：微调概率（Pitch Adjusting Rate, PAR）是新和声产生过程中，对新产生的音调进行微调的概率。

微调带宽：微调带宽（bandwidth, BW）是新音调在微调概率下进行微调时，微调的范围。

(3) 和声搜索算法步骤

Step1：定义问题以及初始化参数：确定待解决问题的数学形式，初始化 HMS、HMCR、PAR 和 BW 等参数。

Step2：初始化和声记忆库：随机生成 HMS 个和声放入和声记忆库，并记录每条和声对应的目标值。和声记忆库形式如下：

$$HM=\begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^{HMS} \end{bmatrix}=\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_n^{HMS} \end{bmatrix}\begin{bmatrix} f(X^1) \\ f(X^2) \\ \vdots \\ f(X^{HMS}) \end{bmatrix} \tag{6-8}$$

Step3：产生新和声：新和声中每个音调（决策变量）有三种产生方式：从和声记忆库中选取、音调微调和随机生成。

从和声记忆库中选取时，音调 x_i' 是从 $x_i^1 \sim x_i^{HMS}$ 中随机选取的一个值；随机生成时，音调 x_i' 是从其所有可能的取值中随机生成的一个值。具体如下：

$$x_i' \leftarrow \begin{cases} x_i' \in \left\{x_i^1, x_i^2, ..., x_i^{HMS}\right\} & rand1 \le HMCR \\ x_i' \in R_i & rand1 > HMCR \end{cases} \tag{6-9}$$

其中，$rand1$ 是 0~1 之间的随机数，R_i 是 x_i' 所有可能取值的集合。

当新音调 x_i' 取值后，以 PAR 的概率对其进行微调，具体如下：

$$x_i' \leftarrow x_i' \pm rand() * BW \qquad rand2 \le PAR \tag{6-10}$$

其中，$rand()$ 和 $rand2$ 为 0~1 之间的随机数。

Step4：更新和声记忆库：若新产生和声的目标函数值比 HM 中最差和声的目标函数值好，则用新和声替换 HM 中最差和声；否则舍弃新和声。进入 Step5。

Step5：检查算法终止条件：判断算法是否已经获得满足条件的最优解或者已达到最大迭代次数，若是，则算法终止，否则重复 Step3-Step4。

和声搜索算法流程图如下：

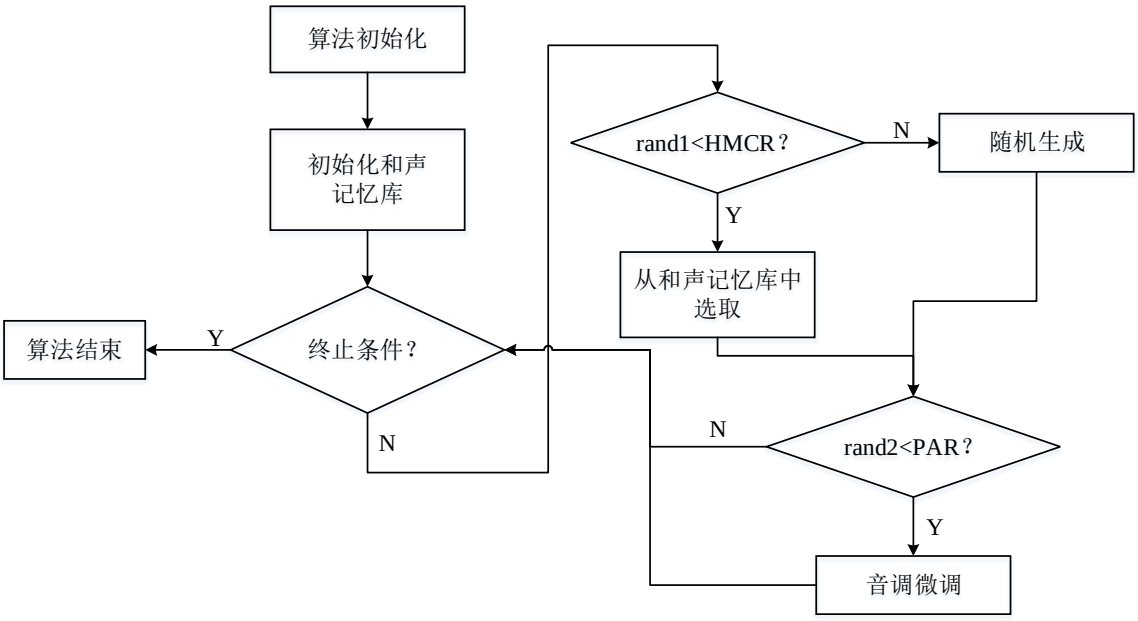

图 6-2 和声搜索算法流程图

6.4.2. 算法关键点设计

(1) 和声编码方式

算法的编码方式建立了问题解空间到编码空间的映射关系，确定了算法的搜索范围。同时，一种合适的和声编码方式可以在新和声产生后带来一个可行的调度方案，这也是和声编码方式设计的参考准则之一。

本文采用基于操作变量的和声编码方式，其中，和声中每一个音调代表一个操作变量，所有音调形成的和声表示了该问题一组可行的操作变量组合，具体如下：

$$harmony_i = [x_{i1}, x_{i2}, ..., x_{i16}] \tag{6-11}$$

其中，$harmony_i$ 表示和声记忆库中第 i 条和声，x_{ij} 表示第 j 个音调，即第 j 个操作变量。

(2) 和声解码方式

解码方式是对每条和声求其适应度值的过程。首先证明辛烷值（RON）损失降幅与优化后产品辛烷值为正向线性指标，证明过程如下：

证明：

$$\begin{aligned} Z &= \left| \frac{\Delta RON_{before} - \Delta RON_{after}}{\Delta RON_{before}} \right| \\ &= \left| \frac{(RON - RON_{before}) - (RON - RON_{after})}{\Delta RON_{before}} \right| \\ &= \left| \frac{RON_{after}}{\Delta RON} - \frac{RON_{before}}{\Delta RON} \right| \\ &= \left| k * RON_{after} + b \right| \end{aligned} \tag{6-12}$$

其中，Z 为辛烷值（RON）损失降幅，RON_{after} 为优化后的操作参数下产品辛烷值的预

测值，k、b 为常数（其他变量含义见符号系统表 2-1）。

由此，将自适应和声搜索算法的适应度函数表示为当前和声下产品辛烷值的预测值，如下所示：

$$h(harmony_i) = f(harmony_i) = f([x_{i1}, x_{i2}, ..., x_{i16}]) \tag{6-13}$$

其中，$h(harmony_i)$为算法的适应度函数，$f(harmony_i)$为问题三求出的产品辛烷值预测模型。

(3) 参数自适应调整

在和声搜索算法中，音调微调概率(Pitch Adjusting Rate, PAR)决定了算法局部搜索能力的大小。在算法迭代初期，和声记忆库中和声解的质量较差，算法需要较强的全局搜索能力，而对局部搜索能力的需求不高；在算法迭代后期，和声记忆库中和声解质量较好，算法需要较强的局部搜索能力，而对全局搜索能力的需求不高。由此，本文考虑在迭代过程中对 PAR 值的大小进行自适应动态调整，调整方式如下所示：

$$PAR_k = PAR_{\min} + \frac{PAR_{\max} - PAR_{min}}{MaxIter} \bullet k \tag{6-14}$$

其中，PAR_k 是第 k 次迭代时的 PAR 值，$PAR_{\max}$ 是 PAR 的最大值，$PAR_{\min}$ 是 PAR 的最小值，$MaxIter$ 是最大迭代次数。

6.4.3. 求解结果

采用上述自适应和声搜索算法对 325 个样本的操作变量取值进行优化，得到 325 个样本在各自当前非操作变量取值情况下的最大辛烷值（RON）损失降幅以及相应的操作变量方案（完整结果见附表“问题 4-优化结果.xlsx”），统计各个辛烷值（RON）损失降幅级数的频数和频率，如下表所示：

表 6-2 辛烷值损失降幅级数频数分析表

降幅范围（单位:%）	频数	频率
(60,100]	0	0.0000
(50,60]	11	0.0338
(40,50]	105	0.3230
(30,40]	117	0.3600
(20,30]	53	0.1633
(10,20]	23	0.0708
(0,10]	12	0.0368
无效样本	4	0.0123
总计	325	1.0000

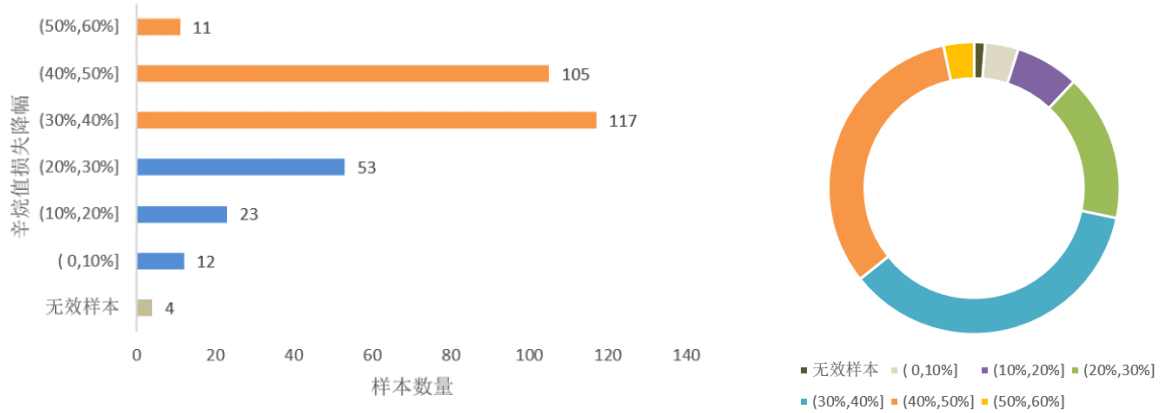

图 6-3 样本优化辛烷值降幅频数分析图

表 6-1 中“无效样本”表示无论操作变量如何取值，优化结果均不满足产品硫含量约束，优化后的产品硫含量均大于 5μg/g。

其中，辛烷值损失降幅大于 30%的样本有 233 个，占比达 71.7%，小部分样本降幅在 30%以下，且大多接近 30%。部分样本损失降幅接近 60%，原因在于存在少量原本辛烷值损失大于 1.7 的样本，具有较大的可优化空间。

另外，存在 4 个样本未能在约束条件下求得较优解，经分析发现，该 4 个样本的初始硫含量较大，都在 310 μ g/g 以上，脱硫难度较大，而这些样本在原始条件下精炼的辛烷值损失仅为 1.0 左右，在所有样本中偏小，说明原始条件下的精炼效果已经属于较优水平，因而具有很大的优化难度。

为了进一步探究精炼过程中操作变量的分布，我们选取降幅大于 30%的 233 个样本数据，对优化后的 16 个主要操作变量取值的分布规律进行了分析，绘制出的频数分布图和拟合的正态分布曲线如图 2 所示。部分变量分布与拟合的正态分布曲线重合度较高，此类变量取值较为集中，说明将该类操作变量调整到正态分布的均值附近，获得较低的辛烷值损失的可能性较大，当然，操作变量之间高度非线性，这个值仅能当作参考。

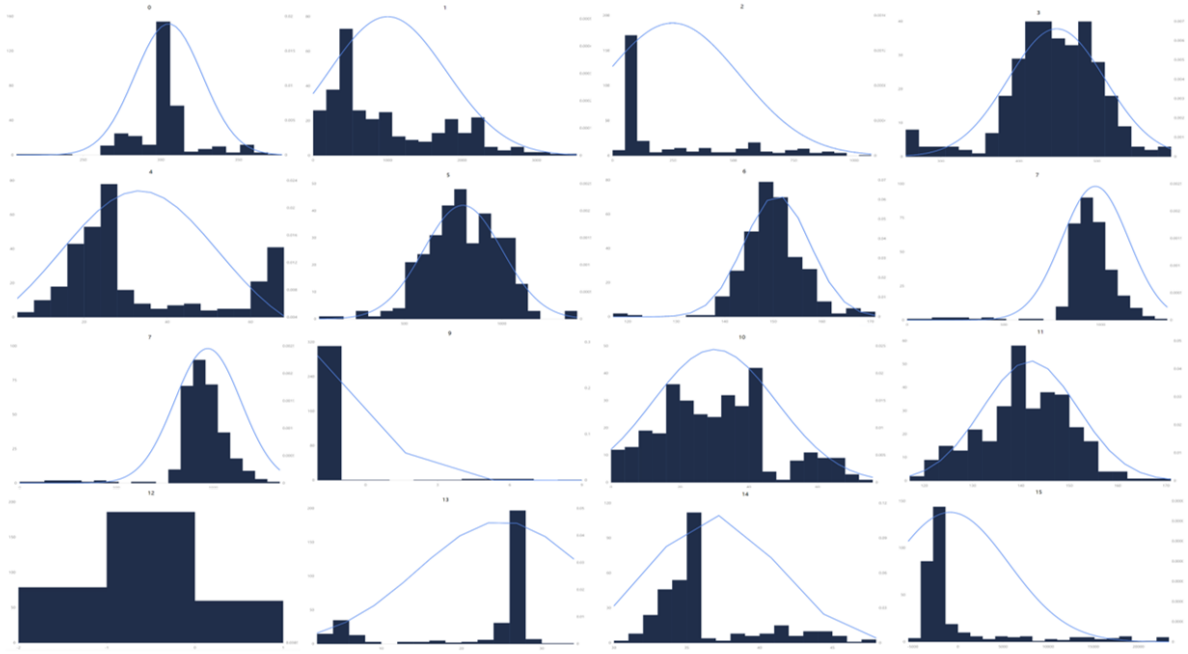

图 6-4 16 个主要操作变量分布分析图

考虑到为工业现场提供更有价值的参考，可根据优化后的操作变量的分布建立一个具有导向性的优化取值范围。为了减小数据中奇异值对范围的影响，可利用拟合的正态分布曲线对范围进行确定。经测试，取每个操作变量正态拟合曲线的(μ-2σ，μ+2σ)作为优化取值范围，各变量具体取值如表 6-3 所示，各变量优化取值范围相对于原始取值范围缩小程度如图 6-5 所示。本文所求得的优化取值范围缩小了操作变量的合理取值区间，能对工业现场操作变量的调整提供更精确的参考和指引。

表 6-3 优化取值范围

名称	还原器温	轻烃出装	净化风进装置	EH-103加热元	K-101B左排气	P105A/B出口总	D104温度	非净化风
最小值	250	300	50	300	10	500	140	820
最大值	310	3000	750	500	65	1000	150	900
名称	加热炉主	D-124液位	ME-104出入口	E-101壳程出	F-101辐射室	K-102B进气	精制汽油	反应器料
最小值	0.1	-2	2	127	-1.8	2	32	-4000
最大值	0.2	3	55	150	0	30	45	17000

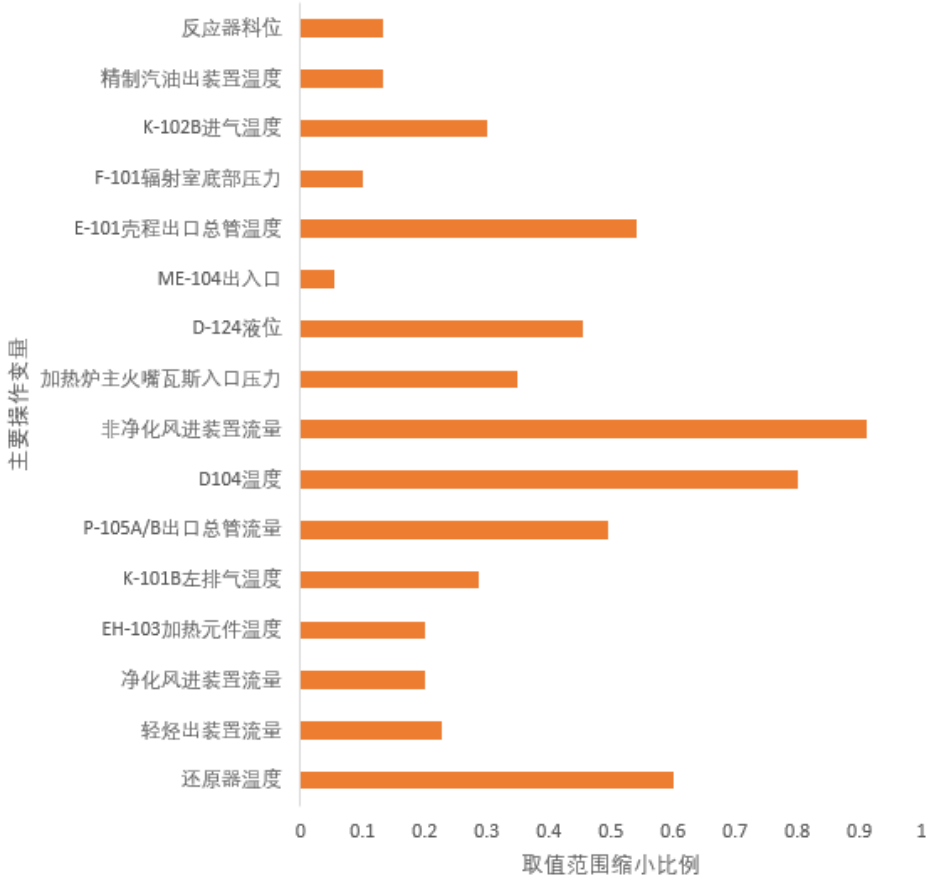

图 6-5 优化取值范围相对于原始取值范围缩小程度

6.5 对比分析

本文将采用基于二分法的启发式算法求解上述线性规划模型，并与和声搜索算法的求解效果进行对比分析。二分法是解决函数优化问题的常用方法，一种适用于本模型的基于二分法的启发式算法描述及步骤如下：

Step1：保持其他主要操作变量取值不变，随机选择一个主要操作变量进行二分法优化，记录过程中的最好解，直到无法继续优化停止；

Step2：判断是否满足算法终止条件，若满足，则转到 step3；否则继续随机选择一个主要操作变量，重复 step1~step2；

Step3：输出算法寻优过程中的最好解。

为验证所使用的和声搜索算法的优化效果，使用一种基于二分法的启发式算法，从求解质量等方面，与其进行对比分析。算法优化效果对比如表 6-4 所示。

表 6-4 算法优化效果对比

	降幅大于 30%样本数量	降幅大于 30%样本占比
和声搜索算法	233	71.7%
启发式算法	126	38.8%

采用 5 个样本作为代表样本，两种算法对数据样本优化的最优解如表 6-4 所示。

表 6-4 最优解对比

样本编号	16	52	140	222	271
和声搜索算法	87.79	89.38	87.90	89.76	89.05
启发式算法	87.02.	88.95	87.53	89.34	88.58

通过上述对比分析可知，基于二分法的启发式算法优化得出降幅大于 30%的样本数量远低于和声搜索算法，进一步通过部分样本的最优解对比可知，和声搜索算法的求解质量更好，对该问题具有的良好的优化效果。

6.6 结果结论

(1) 采用自适应和声搜索算法求解以最大化辛烷值损失降幅为目标的线性规划模型，得到 325 个样本在各自当前非操作变量取值情况下的最大辛烷值（RON）损失降幅以及相应的操作变量方案，并进一步分析了精炼过程中操作变量的分布。

(2) 利用基于二分法的启发式算法与自适应和声搜索算法的求解效果进行对比，可以得到和声搜索算法的求解效果更好，求得解的质量更高。

7. 问题五： 模型可视化展示

7.1. 问题分析

针对问题五，要求对模型进行可视化展示。实际生产中，由于工业装置的平稳性要求，优化后的主要操作变量往往只能逐步调整到位。考虑到问题特性和汽油精制过程的复杂

性，拟从三个层面进行本文模型的可视化展示：一、产品性质层，该层中主要展示主要操作变量按次优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹；二、操作方案层，该层主要展示各个主要操作变量随调整次数的变化轨迹；三、内在关联层，该层主要展示产品性质中汽油辛烷值和硫含量随操作变量的变化轨迹。本章以 133 号样本为主要展示对象，思路流程图如下：

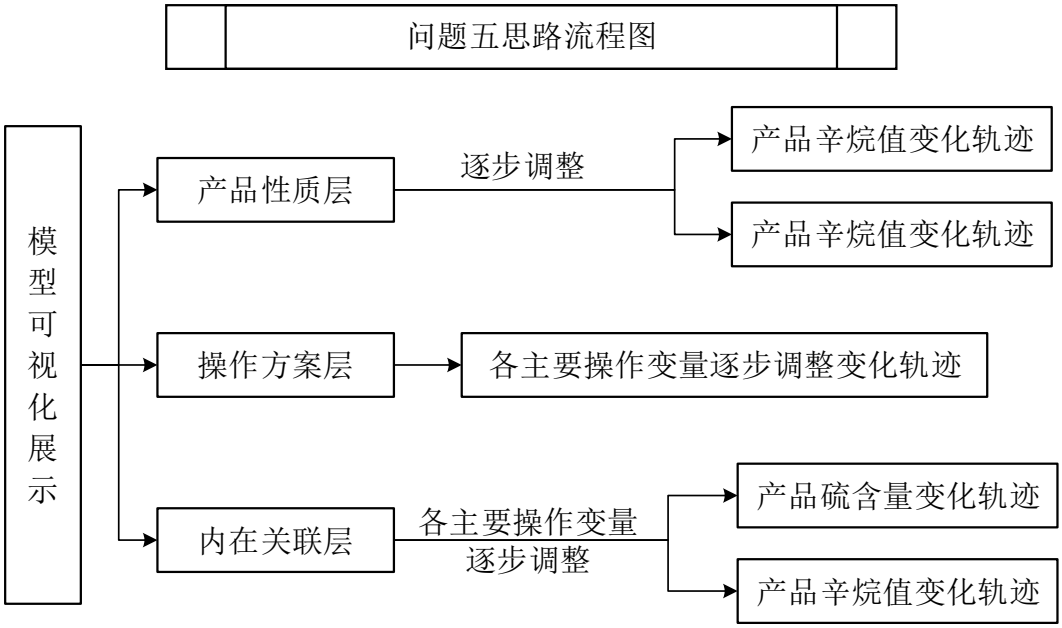

图 7-1 问题五思路流程图

7.2. 产品性能层可视化

该层中主要展示主要操作变量按次优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹，133 号样本该层的可视化展示如图 7-2 所示：

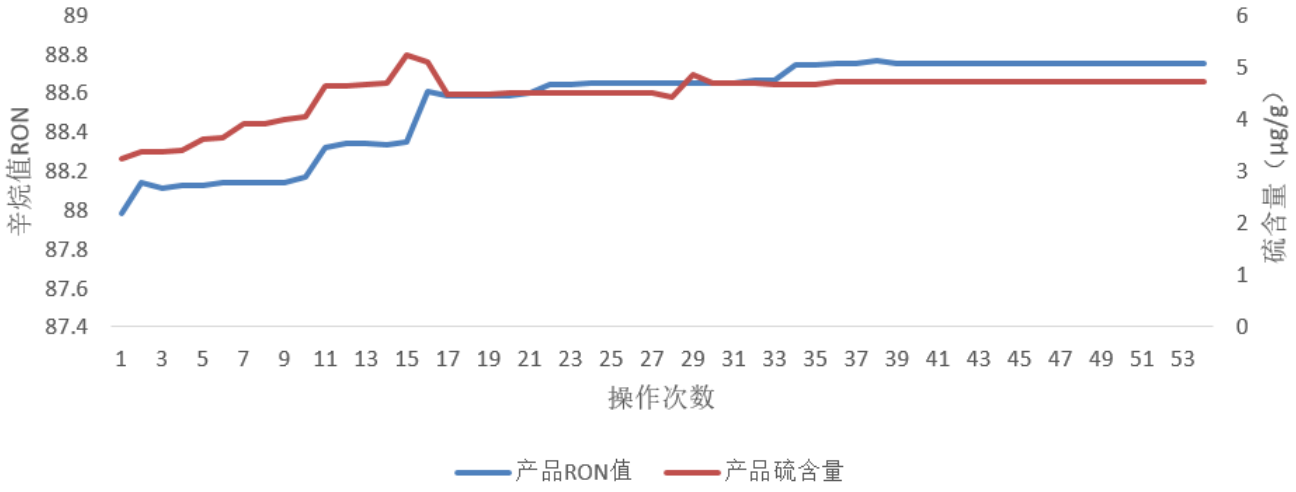

图 7-2 辛烷值和硫含量随调整次数的整体变化轨迹

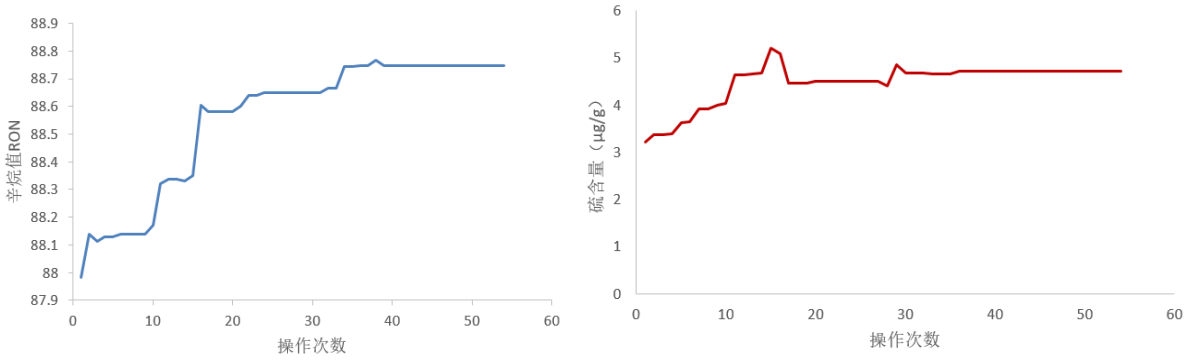

图 7-3 辛烷值和硫含量随调整次数的独立变化轨迹

7.3. 操作方案层可视化

该层主要展示各个主要操作变量随调整次数的变化轨迹，便于直观比较不同主要操作变量的调整幅度大小，133 号样本该层的可视化展示部分如图 7-4 所示：

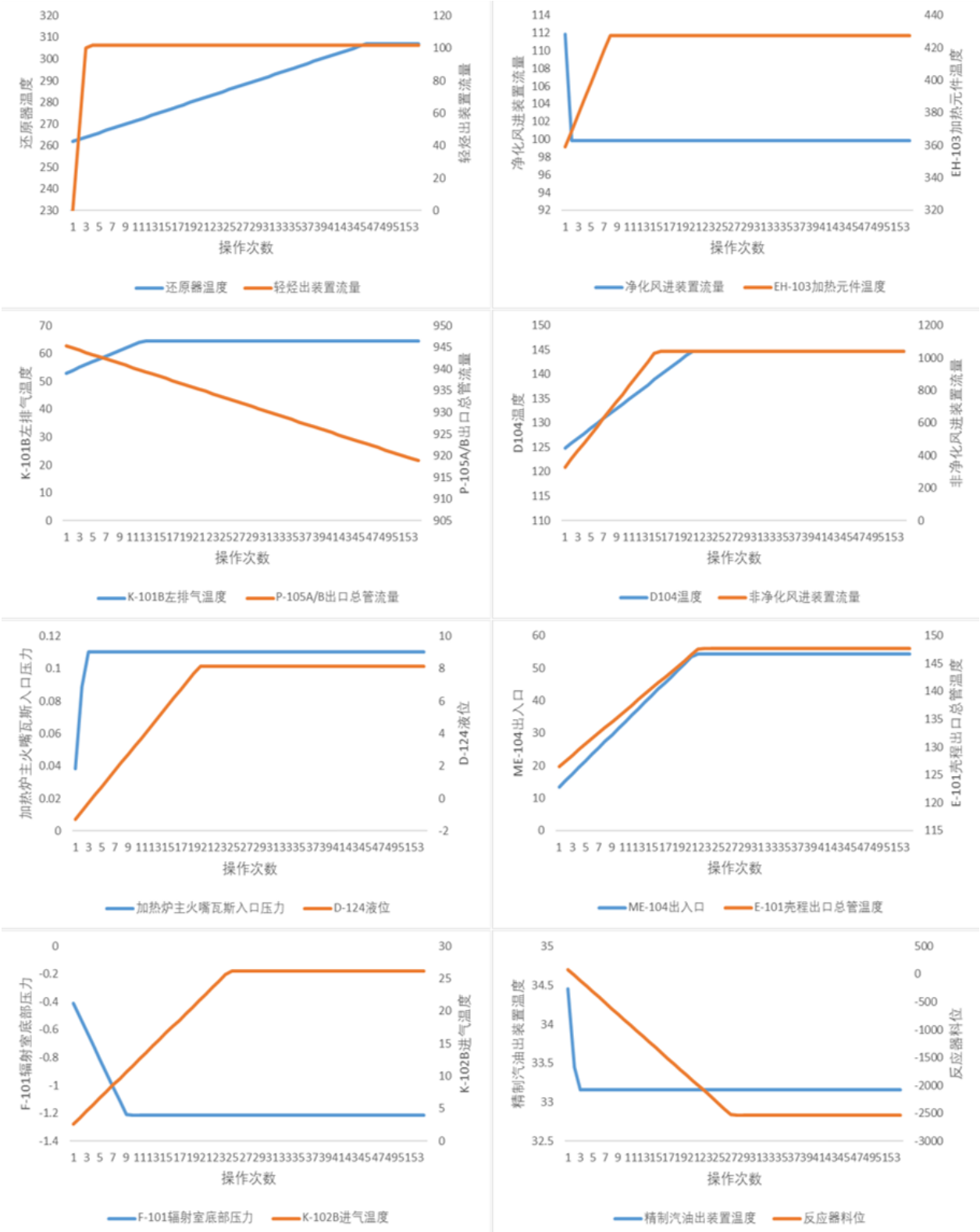

图 7-4 主要操作变量随调整次数的变化轨迹

7.4. 内在关联层可视化

该层主要展示产品性质中汽油辛烷值和硫含量随操作变量的变化轨迹，便于直观发现产品性质中汽油辛烷值和硫含量与主要操作变量之间的联系，133 号样本该层的可视化展示部分如下图所示：

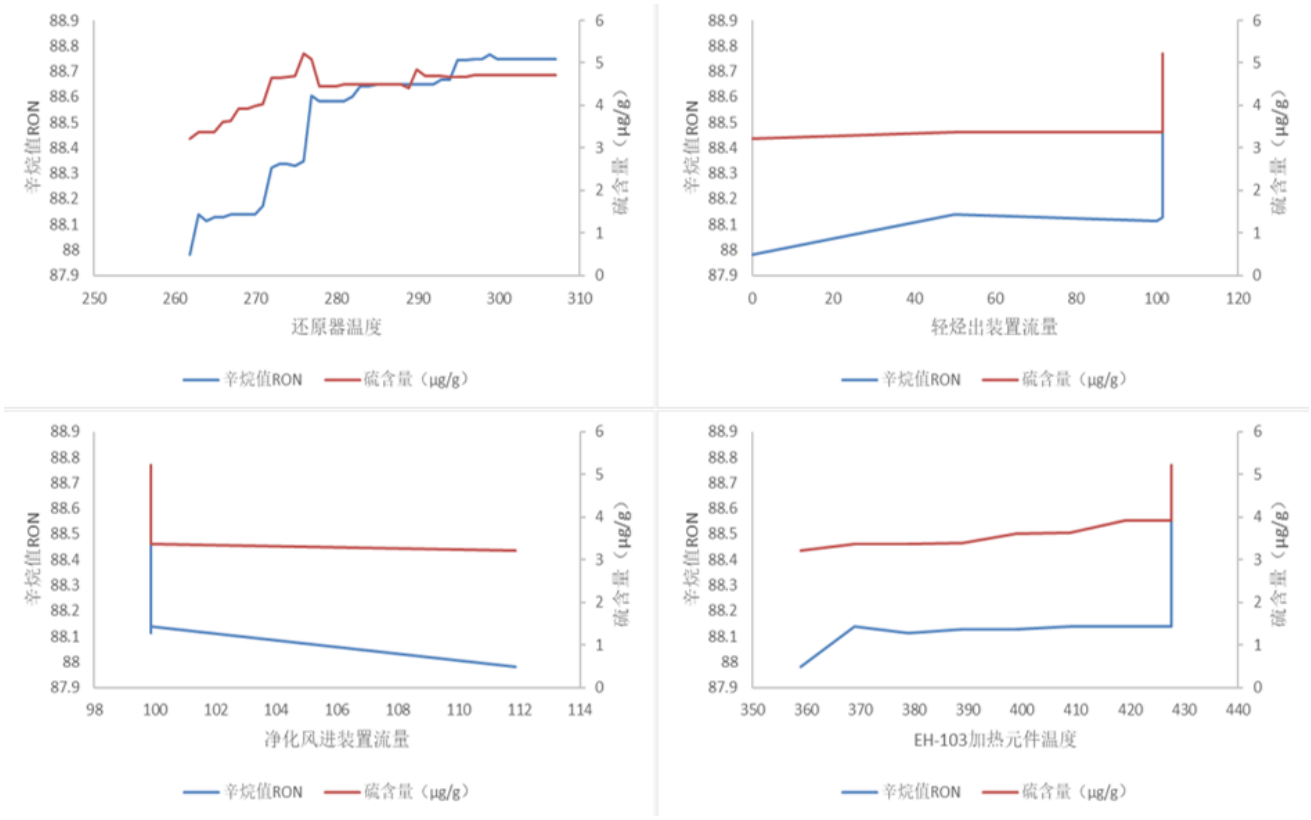

图 7-5 辛烷值和硫含量与主要操作变量变化关系图一

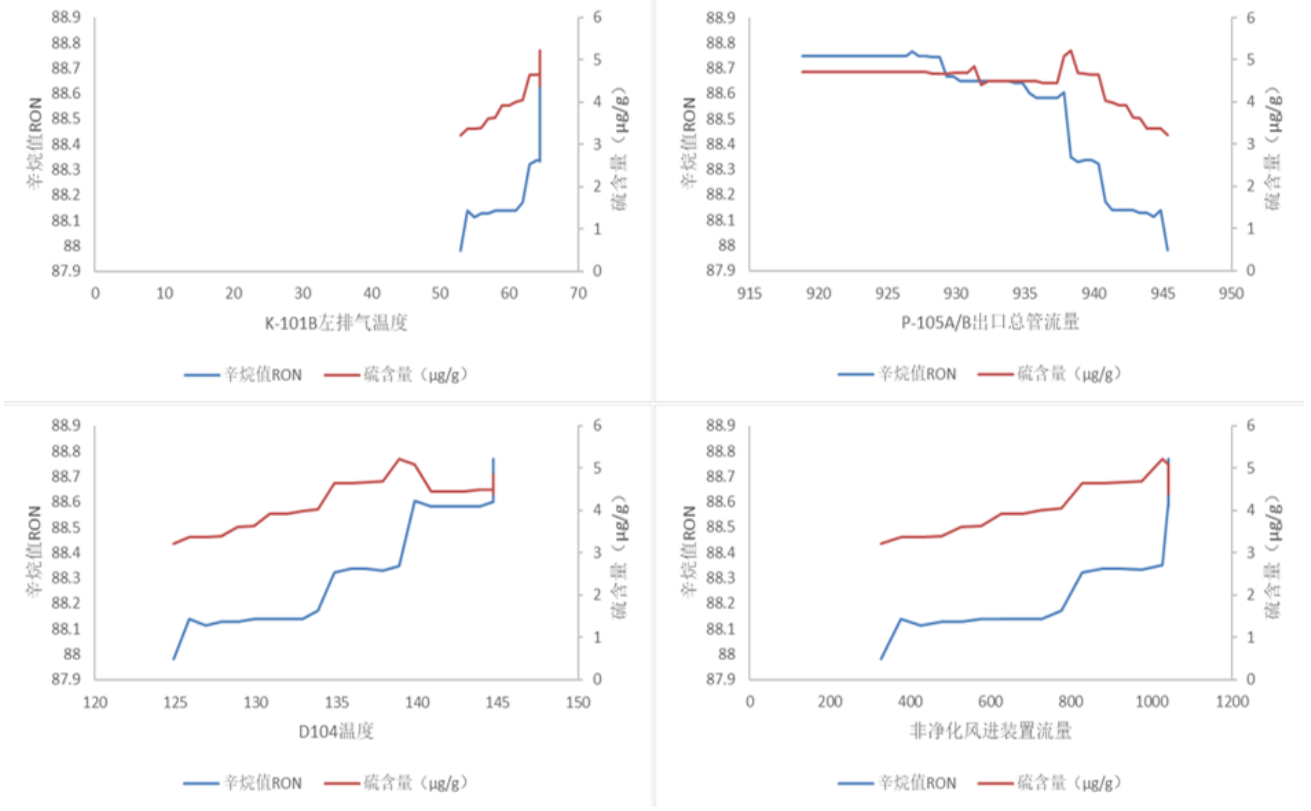

图 7-6 辛烷值和硫含量与主要操作变量变化关系图二

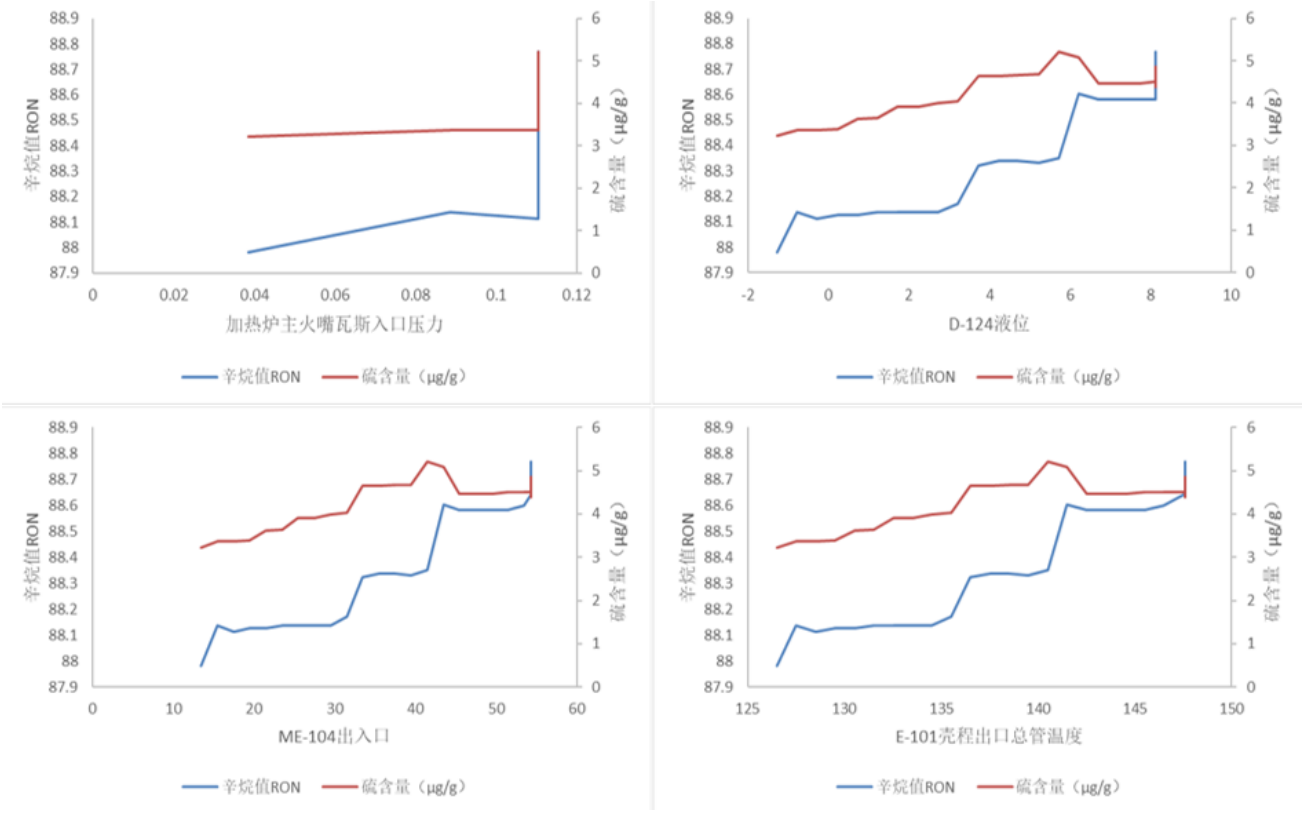

图 7-7 辛烷值和硫含量与主要操作变量变化关系图三

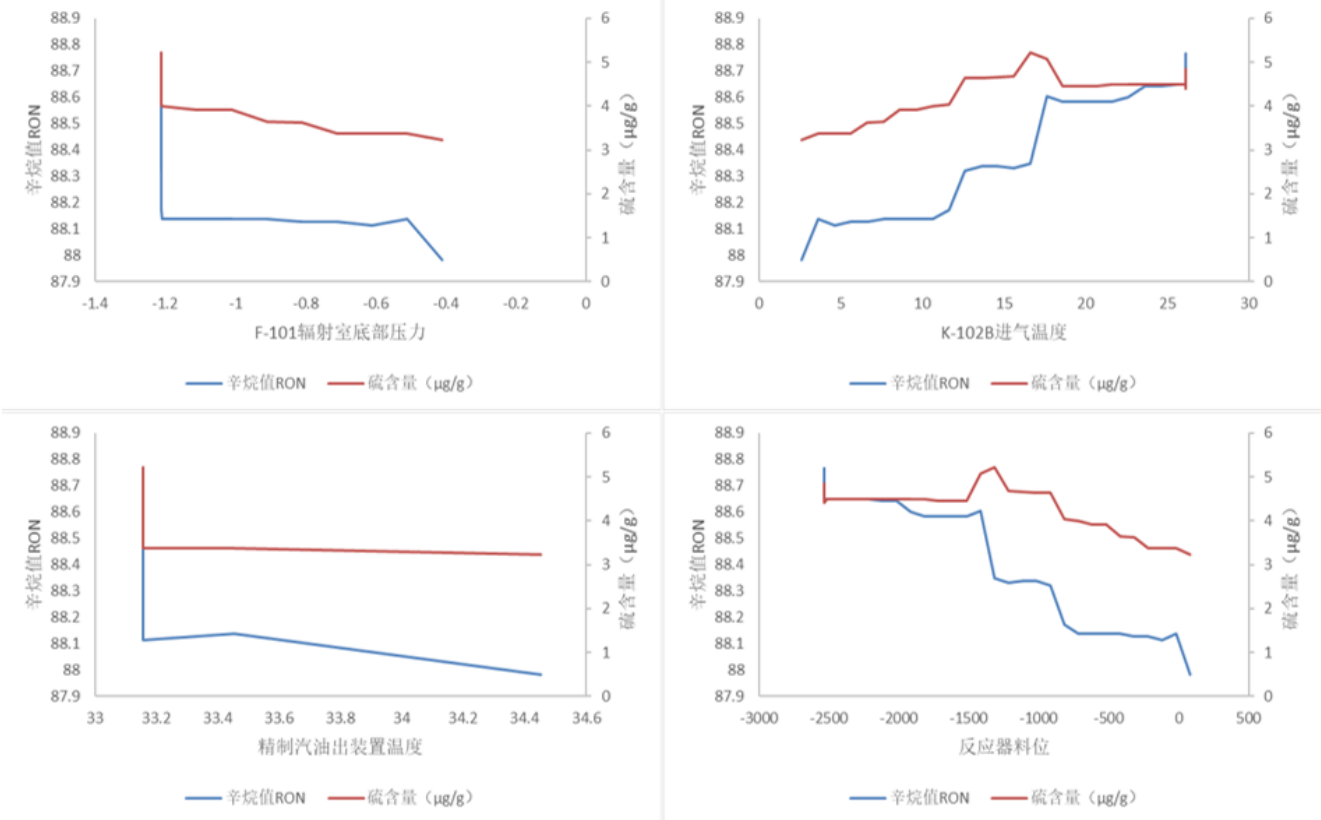

图 7-8 辛烷值和硫含量与主要操作变量变化关系图四

8. 模型的评价与推广

8.1. 模型的评价

8.1.1. 模型的优点

(1) 本文在考虑解决问题时，采用了多个模型。如在针对问题三建立产品辛烷值的预测模型时，本文分别建立了基于多元线性回归的预测模型、基于随机森林的预测模型以及基于 GBR 的预测模型，并且针对不同的模型进行交叉验证与评估，极大地保证了预测结果的合理性和完善性。

(2) 本文在研究问题时考虑较为全面。如针对问题三进行了敏感性对比，针对问题四进行了算法的对比法分析。敏感性分析与对比分析不但保证了文章的严谨性，特别是针对工业场景进行应用，更需要对所建立的模型以及结果进行分析，保证现场工作的可靠性，满足实际生产应用需求。

8.1.2. 模型的缺点

(1) 本文虽通过多种方法得到了较准确的模型，但由于样本数量的限制，所得到的模型对实际汽油精练过程概况程度有限，但本文所提方法对解决该问题效果显著，若能获得更多的数据样本，能得到更具准确性和实用性的模型。

(2) 由于专业的限制，本文对汽油精炼过程中各种变量的化学本质并未作深入分析，大多数工作是基于数据的，若能融合专业相关知识，能对该问题进行更加透彻的分析。

8.2. 模型的推广

在后续研究中，为保证同时兼顾数据的深度与广度，需要提高样本数量，对数据集进行多次训练，不断优化模型，提升模型预测精度。对于本文筛选出的变量，并未分析其在模型中的具体表现，因此相关石化从业人员可考虑使用本文的模型进行进一步的内在联系模式的探究。

本文主要是研究辛烷值在相关主要变量的作用下的预测与优化，但是建立的模型也可以预测与优化汽油精制过程中其他一些产品的产值，即本文的模型具有一定的推广价值。

参考文献

[1] Natekin A, Knoll A. Gradient boosting machines, a tutorial[J]. Frontiers in Neurorobotics. 2013, 7(UNSP 21).

[2] Pan Y, Chen S, Qiao F, Ukkusuri S V, Tang K. Estimation of real-driving emissions for buses fueled with liquefied natural gas based on gradient boosted regression trees[J]. Science of the Total Environment. 2019, 660: 741-750.

[3] Torres-Barran A, Alonso A, Dorronsoro J R. Regression tree ensembles for wind energy and solar radiation prediction[J]. Neurocomputing. 2018, 326: 151-160.

[4] Wang F, Mamo T. Gradient boosted regression model for the degradation analysis of prismatic cells[J]. Computers & Industrial Engineering. 2020, 144(106494).

附录

问题 1　C＋＋程序	数据预处理

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
#include <sstream>
#include <cmath>
#include <cstdlib>
#include <read_data.h>
#include <trans_record.h>
#include <map>
using namespace std;
string infile = "操作变量-处理后.txt";
vector<string>    operate_name;
vector<vector<double>> operate_val(326);
int odd_sum[354];
map<int, int> mymap;
int cnt = 0;
void readdata_badval()
{
	ifstream myfile(infile);
	string line;
	if (!myfile.is_open())
	{
		cout << "未成功打开文件" << endl;
	}
	string s_s;
	getline(myfile, line);
	stringstream sin(line);
	for (int i = 0; i < 354; i++)
	{
		sin >> s_s;
		operate_name.emplace_back(s_s);
	}
	double d_d;
	cnt = 1;
```

```
	while (myfile && cnt < 326)
	{
		getline(myfile, line);
		stringstream sin(line);
		//cout << line << endl;
		for (int i = 0; i < 354; i++)
		{
			sin >> d_d;
			operate_val[cnt].emplace_back(d_d);
		}
		++cnt;
	}
}
string infilerange = "变量范围.txt";
vector<vector<double>> valrange(354, vector<double>(2));
void readdata_range()
{
	ifstream myfile(infilerange);
	string line;
	if (!myfile.is_open())
	{
		cout << "未成功打开文件" << endl;
	}
	string s_s;
	char c;

		while (myfile && cnt < 354)
		{
			getline(myfile, line);
			stringstream sin(line);
			sin >> valrange[cnt][0] >> c >> valrange[cnt][1];
			++cnt;
		}

}

vector<vector<bool>> operate_P(326, vector<bool>(354,true));//是否正常
vector<bool> odd_col(354,false);
int main()
{
	readdata_range();
	readdata_badval();
	for (int j = 0; j < 354; j++)
	{
```

```
            for (int i = 1; i < 326; i++)
            {
                if (operate_val[i][j] < valrange[j][0] || operate_val[i][j] > valrange[j][1])
                {
                    odd_sum[j]++;
                    operate_P[i][j] = false;
                }
            }
        }
        int allnum = 0;
        int oddnum = 0;
        int singleoddnum = 0;

        for (int j = 0; j < 354; j++)
        {
            if (odd_sum[j] != 0)
            {
                cout << operate_name[j]<<" 编号为  "<< j << "  的缺失数据个数为 " << odd_sum[j];
                allnum += odd_sum[j];
                oddnum++;
                if ((float)odd_sum[j] / 325 >= 0.3)
                {
                    singleoddnum++;
                    odd_col[j] = true;
                    cout << " 且缺失率大于 0.3, 为 " << (float)odd_sum[j] / 325;
                }
                cout << endl;
            }
        }
        cout << endl;
        cout << endl;
        cout << "共有 " << oddnum << " 个位点存在缺失数据 " << endl;
        cout << "总共缺失 " << allnum << " 条数据 " << endl;
        cout << "共有 " << singleoddnum << " 个位点缺失率大于 0.3 " << endl;


        for (int j = 0; j < 354; j++)
        {
            if (odd_col[j])continue;
            ok_num = 0;
            for (int i = 1; i < 326; i++)
            {
                if (operate_val[i][j] != 0)
```

```
                {
                    average_val[j] += operate_val[i][j];
                    ok_num++;
                }
            }
            average_val[j] /= ok_num;
        }

        for (int j = 0; j < 354; j++)
        {
            if (odd_col[j])continue;
            for (int i = 1; i < 326; i++)
            {
                if (operate_P[i][j])
                {
                    rem_error_p[i][j] = operate_val[i][j] - average_val[j];
                    rem_error_p[i][j] = rem_error_p[i][j] > 0 ? rem_error_p[i][j] :
-rem_error_p[i][j];
                    rem_error_2_sum[j] += rem_error_p[i][j] * rem_error_p[i][j];
                }
            }
            sigema[j] = sqrt(rem_error_2_sum[j] / 324);
        }

        for (int j = 0; j < 354; j++)
        {
            if (odd_col[j])continue;
            for (int i = 1; i < 326; i++)
            {
                if (operate_P[i][j])
                {
                    if (rem_error_p[i][j] > 3 * sigema[j])
                    {
                        operate_P[i][j] = false;
                        cout << operate_name[j] << "    行：" << i << endl;
                        rem_error_num++;
                    }
                }
            }
        }

        cout << "坏点数量：" << rem_error_num << endl;
```

```
vector<double> aver_col(354,0);
int colrightnum;
for (int j = 0; j < 354; j++)
{
	colrightnum = 0;
	for (int i = 1; i < 326; i++)
	{
		if (odd_col[j]) continue;//fs << '#'<<' ';
		else if (operate_P[i][j])
		{
			aver_col[j] += operate_val[i][j];
			colrightnum++;
		}
	}
	aver_col[j] /= colrightnum;
}

ofstream fs;

fs.open("附件 1 补值.txt");

if (!fs) return 0;
for (int i = 1; i < 326; i++)
{
	for (int j = 0; j < 354; j++)
	{
		if (odd_col[j]) continue;//fs << '#'<<' ';
		else if(!operate_P[i][j]) fs << aver_col[j] << ' ';
		else fs << operate_val[i][j] << ' ';
	}
	fs << endl;
}

system("pause");
return 0;
```

问题 3　Python 程序	产品辛烷值预测

```
import numpy as np
import pandas as pd
from sklearn.linear_model import BayesianRidge, LinearRegression, ElasticNet
from sklearn.svm import SVR
from sklearn.ensemble.gradient_boosting import GradientBoostingRegressor  # 集成算法
from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.model_selection import cross_val_score      # 交叉验证
from sklearn.metrics import explained_variance_score, mean_absolute_error, mean_squared_error, r2_score
from sklearn.externals import joblib
import matplotlib.pyplot as plt
import seaborn as sns
#%matplotlib inline

# 数据导入
#1.数据导入
train_data =pd.read_excel('RFR_train.xlsx',index_col=0)
train_data_1 =pd.read_excel('RFR_train.xlsx',index_col=0)
y_train = train_data.pop("P_RON")
# 设置交叉验证次数
n_folds = 5

# 普通线性回归
# 通过加入一个参数 kind='reg'，seaborn 可以添加一条最佳拟合直线和 95%的置信带。
sns.pairplot(train_data_1,  x_vars=['RON','S-ZORB.TC_2801.PV','S-ZORB.PDT_1004.DACA'], y_vars='P_RON', size=7, aspect=0.8,kind = 'reg')
plt.savefig("pairplot.jpg")
plt.show()
lr_model = LinearRegression()
joblib.dump(lr_model,'saved_lr_model')



lr_model.fit(train_data,y_train)
a    = lr_model.intercept_#截距

b = lr_model.coef_#回归系数

# 梯度增强回归模型对象
gbr_model = GradientBoostingRegressor()
joblib.dump(gbr_model,'saved_gbr_model')

#随机森林回归模型
rfr_model = RandomForestRegressor()
joblib.dump(rfr_model,'saved_rfr_model')
# 不同模型的名称列表
model_names = ['LinearRegression', 'GBR','RandomForestRegressor']
# 不同回归模型
model_dic = [br_model, lr_model, etc_model, svr_model, gbr_model,rfr_model]
# 交叉验证结果
cv_score_list = []
```

```
# 各个回归模型预测的 y 值列表
pre_y_list = []

# 读出每个回归模型对象
for model in model_dic:
    # 将每个回归模型导入交叉检验
    scores = cross_val_score(model, train_data, y_train, cv=n_folds)
    # 将交叉检验结果存入结果列表
    cv_score_list.append(scores)
    # 将回归训练中得到的预测 y 存入列表
    pre_y_list.append(model.fit(train_data,  y_train).predict(train_data))
### 模型效果指标评估 ###
# 获取样本量，特征数
n_sample, n_feature = train_data.shape
# 回归评估指标对象列表
model_metrics_name = [explained_variance_score, mean_absolute_error, mean_squared_error, r2_score]
# 回归评估指标列表
model_metrics_list = []
# 循环每个模型的预测结果
for pre_y in pre_y_list:
    # 临时结果列表
    tmp_list = []
    # 循环每个指标对象
    for mdl in model_metrics_name:
        # 计算每个回归指标结果
        tmp_score = mdl(y_train, pre_y)
        # 将结果存入临时列表
        tmp_list.append(tmp_score)
    # 将结果存入回归评估列表
    model_metrics_list.append(tmp_list)
df_score = pd.DataFrame(cv_score_list, index=model_names)# 各个交叉验证的结果
df_met = pd.DataFrame(model_metrics_list, index=model_names, columns=['ev', 'mae', 'mse', 'r2'])
# 各种评估结果


### 可视化 ###
# 创建画布
plt.figure(figsize=(9, 36))
# 颜色列表
color_list = ['r', 'g', 'b']
# 循环结果画图
for i, pre_y in enumerate(pre_y_list):
```

```
    # 子网络
    plt.subplot(6, 1, i+1)
    # 画出原始值的曲线
    plt.plot(np.arange(train_data.shape[0]), y_train, color='k', label='y')
    # 画出各个模型的预测线
    plt.plot(np.arange(train_data.shape[0]), pre_y, color_list[i], label=model_names[i])
    plt.title(model_names[i])
    plt.legend(loc='lower left')
plt.savefig('xxx.png')
plt.show()
```

问题 4　Python 程序	主要变量操作方案优化

```
import pandas as pd
import numpy as np
from sklearn.linear_model import BayesianRidge, LinearRegression, ElasticNet
from sklearn.svm import SVR
from sklearn.ensemble.gradient_boosting import GradientBoostingRegressor  # 集成算法
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score  # 交叉验证
from sklearn.metrics import explained_variance_score, mean_absolute_error,
mean_squared_error, r2_score
from sklearn.externals import joblib
import matplotlib.pyplot as plt
import seaborn as sns
import random

RON_model = joblib.load('saved_gbr_model_RON.pkl')
S_model = joblib.load('saved_gbr_model_S.pkl')
df1 = pd.read_excel("筛选变量最大最小值.xlsx",header=None)
fanwei = np.array(df1.iloc[:,1:3])
nvars=16

def init_har(var_no):
    a = random.uniform(fanwei[var_no][0],fanwei[var_no][1])
    return a

def Func(inf,harmony):
    x1 = np.array(inf).reshape(1,10)
    x2 = np.array(harmony).reshape(1,nvars)
    x = np.concatenate([x1,x2],axis=1)

    s = S_model.predict(x)
    if (s>5):
```

```
            return -1
        ron = RON_model.predict(x)
        return ron


def HS(inf):
        #参数赋值
        #GHS 参数
        HMS = 10
        HMCR = 0.9
        PARmin = 0.05
        PARmax = 0.5
        MaxIteration = 2000

        #初始化 HM
        HM = np.zeros((HMS,nvars))
        for i in range(HMS):
            for j in range(nvars):
                HM[i][j] = init_har(j)

        #计算没条和声的和声适应度
        fitness = np.zeros((HMS,1))
        for i in range(HMS):
            fitness[i] = Func(inf,HM[i])

        #记录每次迭代中的最优
        temp_best = []

        for j in range(MaxIteration):
            x_new = np.zeros(nvars)
            PAR = PARmin+((PARmax-PARmin)/MaxIteration)*j #动态调整 PAR
            #对每一个音符进行操作
            for k in range(nvars):
                if np.random.random() < HMCR:
                    index_temp = np.random.randint(0,HMS)
                    x_new[k] = HM[index_temp][k]
                else:
                    x_new[k] = random.uniform(fanwei[k][0],fanwei[k][1])
                if np.random.random() < PAR:
                    x_new[k]                    =                    x_new[k]                    +
np.random.random()*((fanwei[k][1]-fanwei[k][0])/10)
                    if(x_new[k]>fanwei[k][1]):
                        x_new[k] = random.uniform(fanwei[k][0],fanwei[k][1])
                    if(x_new[k]<fanwei[k][0]):
```

```
                    x_new[k] = random.uniform(fanwei[k][0],fanwei[k][1])


            #计算当代新和声适应度
            y_temp = Func(inf,x_new)
            #更新
            index_temp = np.where(fitness==np.min(fitness))[0][0]
            if y_temp > fitness[index_temp]:
                fitness[index_temp] = y_temp
                HM[index_temp] = x_new
            #记录当代最优并更新全局最优
            temp_best.append(np.max(fitness))

        goal_best = np.max(fitness)
        index_temp1 = np.where(fitness==np.max(fitness))[0][0]
        queue_best = HM[index_temp1]
        return goal_best,queue_best



if __name__=='__main__':
        df = pd.read_excel("变量筛选结果.xlsx",index_col=0)

        df_x = df.iloc[:,0:26]

        df_y = df.iloc[:,26] #产品初始辛烷值
        df_Inf = df_x.iloc[:,0:10]
        df_RON1 = df_x.iloc[:,1] #原料辛烷值

        RON0=np.array(df_y)#产品优化前辛烷值
        RON1=np.array(df_RON1)#原料辛烷值
        INF = np.array(df_Inf)

        #HS(INF[0])
        y_best = np.zeros((325,1))
        fudu = np.zeros((325,1))
        bianliang=[]
        for i in range(325):
            y_best[i],temp = HS(INF[i])
            bianliang.append(temp)
            print(str(i)+"\n")

        count = [];
        for i in range(325):
```

```
            x0 = RON1[i] - RON0[i]
            x1 = RON1[i] - y_best[i]
            if (x1<0):
                  count.append(i)
            fudu[i] = ((x0 - x1)/x0) *100

      res= np.concatenate([y_best,fudu],axis=1)

      df_res = pd.DataFrame(res)
      df_bl = pd.DataFrame(bianliang)

      writer = pd.ExcelWriter('Q4_result.xlsx')        # 写入 Excel 文件
      df_res.to_excel(writer, 'Sheet1', float_format='%.5f')        # ‘page_1’是写入 excel 的
sheet 名
      df_bl.to_excel(writer, 'Sheet2', float_format='%.5f')
      writer.save()
      writer.close()
```

问题 5　Python 程序	可视化展示

```
import numpy as np
import pandas as pd
from sklearn.linear_model import BayesianRidge, LinearRegression, ElasticNet
from sklearn.svm import SVR
from sklearn.ensemble.gradient_boosting import GradientBoostingRegressor  # 集成算法
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score      # 交叉验证
from sklearn.metrics import explained_variance_score, mean_absolute_error,
mean_squared_error, r2_score
from sklearn.externals import joblib
import matplotlib.pyplot as plt
import seaborn as sns
#%matplotlib inline


time = np.loadtxt("time.txt",dtype=int)
before = np.loadtxt("before.txt")
after = np.loadtxt("after.txt")
deta = np.loadtxt("deta.txt")

k = max(time)
l = len(before)
pro = np.zeros((k,l))
```

```
pro[0,:] = before
for i in range(1,k):
    for j in range(l):
        if before[j]<after[j]:
            m = before[j] + deta[j]
            if m>after[j]:
                pro[i,j] = after[j]
            else:
                pro[i,j] = m
            before[j] = m
        elif before[j]>after[j]:
            n = before[j] - deta[j]
            if after[j]<n:
                pro[i,j] = n
            else:
                pro[i,j] = after[j]
            before[j] = n

RON_model = joblib.load('saved_gbr_model_RON.pkl')
S_model = joblib.load('saved_gbr_model_S.pkl')
test_data =pd.read_excel('test.xlsx',index_col=0)
S_test = S_model.predict(test_data)
RON_test = RON_model.predict(test_data)
```